# APT HOWTO

Gustavo Noronha Silva <kov@debian.org>
Deutsche Übersetzung: David Spreen <netzwurm@debian.org>

2.0.2 - Oktober 2006

## Zusammenfassung

Dieses Dokument soll dem Benutzer ein gutes Verständnis für die Arbeitsweise des Debian-Paketmanagement-Werkzeugs APT liefern. Ziel ist es, das Leben für neue Debian-Benutzer zu erleichtern und denen zu helfen, die ihr Verständnis für die Administration dieses Systems vertiefen wollen. Es wurde für das Debian-Projekt geschaffen, um den verfügbaren Support für Benutzer dieser Distribution zu erweitern.

## Copyright-Hinweis

# Inhaltsverzeichnis

**1 Zur Deutschen Übersetzung** **1**

1.1 Vorwort . . . . . 1

1.2 Glossar . . . . . 1

**2 Einführung** **3**

2.1 Grundlegende Begriffe und Konzepte . . . . . 3

**3 Einführung in das Debian-Archiv** **5**

3.1 Einführung in Debian-Suiten . . . . . 5

3.2 Debians Sektionen . . . . . 5

3.3 APTs Vertrauenskette . . . . . 6

**4 Basis-Konfiguration** **7**

4.1 Die Datei /etc/apt/sources.list . . . . . 7

4.2 Pakete für echt befinden . . . . . 8

4.3 Hinzufügen einer CD-ROM in die Datei sources.list . . . . . 9

4.4 Die APT Konfigurationsdatei . . . . . 10

4.5 APT-Prioritäten für Pakete anpassen . . . . . 10

4.6 APT durch einen Proxy hindurch benutzen . . . . . 12

**5 Paketverwaltung** **13**

5.1 Debians Paketmanager benutzen: »aptitude« . . . . . 13

5.2 Update der Liste der verfügbaren Pakete . . . . . 15

5.3 Nach Paketen suchen . . . . . 16

5.4 Installieren und Reinstallieren von Paketen . . . . . 16

5.5 Manuell heruntergeladene oder lokal erzeugte Debian-Pakete installieren . . . . 19
5.6 Pakete entfernen . . . . 20
5.7 Aktualisieren von Paketen . . . . 21
5.8 Aktualisierung auf eine neue Debian-Version . . . . 21
5.9 Pakete auf »halten« setzen . . . . 22
5.10 Ungenutzte Paket-Dateien entfernen: »clean« und »autoclean« . . . . 23

**6 Aus den Quellen bauen** **25**
6.1 Herunterladen von Quellpaketen . . . . 25
6.2 Für das Kompilieren eines Quellpaketes notwendige Pakete . . . . 26
6.3 Ein Paket zur Fehlersuche bauen . . . . 27
6.4 Anpassen von Optionen des Paket-Baus . . . . 27
6.5 Aber heh, ich will den Debian-Kram nicht benutzen! . . . . 28

**7 Informationen über Pakete bekommen** **33**
7.1 Paketnamen entdecken . . . . 33
7.2 Paketnamen mit dpkg finden . . . . 36
7.3 Herausfinden, zu welchem Paket eine Datei gehört . . . . 37
7.4 Informationen über Pakete im Web . . . . 37

**8 Tricks und Techniken** **39**
8.1 Wie Pakete »nach Bedarf« installieren . . . . 39
8.2 Pakete zwischenspeichern . . . . 39
8.3 Entscheidung - Welcher Spiegel ist der beste für die sources.list: netselect, netselect-apt . . . . 41
8.4 Entfernen von unbenutzten locale-Dateien: localepurge . . . . 42
8.5 Wie über Änderungen in Paketen informiert bleiben . . . . 42

**9 Auf der anderen Seite: ein Repository erzeugen** **45**

**10 Alternative Benutzerschnittstellen zur Paketverwaltung** **47**
10.1 Synaptic . . . . 47
10.2 GNOME-APT . . . . 47

**11 Der Umgang mit Fehlern** **49**

11.1 Häufige Fehler . . . . . . . . . . 49

11.2 Wo gibt es Hilfe? . . . . . . . . . . 50

**12 Welche Distributionen unterstützen APT?** **51**

**13 Danksagungen** **53**

**14 Neue Versionen dieses Dokuments** **55**

# Kapitel 1

# Zur Deutschen Übersetzung

## 1.1 Vorwort

Der Text dieser Übersetzung hat das Ziel, sich möglichst eng an der englischen Übersetzung zu orientieren. Leider lassen sich einige Fachbegriffe schlecht übersetzen. Diese Begriffe habe ich versucht, entweder sinnvoll einzudeutschen oder als englisches Wort zu übernehmen. Alle diese Begriffe werden im untenstehenden Glossar kurz erläutert.

## 1.2 Glossar

Die Begriffe sind hier in der Reihenfolge aufgelistet, in der sie im Text auftauchen.

- *Kompilieren* (eingedeutscht): Das Übersetzen des Programmquelltextes in ein für den Computer ausführbares Programm.
- *Paket* (eingedeutscht): Hier Sammlung von Dateien, die meist ein vorkompiliertes und konfiguriertes Programm enthält, welches sich mit einem Paketverwaltungsprogramm installieren lässt.
- *Portieren* (eingedeutscht): Das Verändern eines Programms, um es auf einem anderen System oder einer anderen Architektur benutzen zu können.
- *Update*: Aktualisieren von Informationen oder Software, um sie auf dem neusten Stand zu halten, wobei Paket-Updates bei APT »Upgrade« heißen. Verwirrenderweise heißen Paketupgrades aufgrund von Sicherheitslücken bei Debian allerdings »Sicherheits-Updates«.
- *Spiegel*: Eine exakte Kopie eines Web- oder FTP-Servers, die dazu dient, den Originalserver zu entlasten. Die Liste der Debian-Mirrors gibt es auf http://www.debian.org/distrib/ftplist.
- *Upgrade*: Sehen Sie Update. Hierbei handelt es sich um das Aktualisieren von Paketen.

- *Ncurses*: Programmbibliothek, um auf der Konsole Fenster, Menüs usw. darzustellen.
- *MTA*: Mail Transport Agent - Programm zum Transportieren von E-Mails, oft auch Mailserver.
- *Manpage*: Anleitung für ein Programm, welche auf der Kommandozeile per `man Programmname` verfügbar ist.
- *Lock-Datei*: Datei, deren Existenz anzeigt, daß eine Ressource (z.B. die APT-Datenbank) zur Zeit genutzt wird.

# Kapitel 2

# Einführung

Am Anfang war das .tar.gz. Benutzer mussten jedes Programm, welches sie auf ihren GNU/Linux-Systemen benutzen wollten, selbst kompilieren. Zu Beginn der Entwicklung des Debian-Projekts erachtete man es für notwendig, dass das System eine Methode zum Verwalten der Pakete, die auf dem System installiert sind, enthält. Man gab dieser Methode den Namen `dpkg`. Dadurch war das erste »Paket« auf GNU/Linux geboren, bevor Red Hat sich entschied, ihr eigenes RPM-System zu erschaffen.

Schnell standen die Macher von GNU/Linux vor einem neuen Dilemma. Sie brauchten ein schnelles, praktisches und effizientes Mittel, um Pakete zu installieren, das Abhängigkeiten automatisch behandeln und ihre Konfigurationsdateien während des Aktualisierens berücksichtigen würde. Und wieder war es das Debian-Projekt, das den Weg machte und APT, das »Advanced Packaging Tool« (fortgeschrittenes Paketierungswerkzeug) das Licht der Welt erblicken ließ, welches seitdem von Connectiva auf RPM portiert und von einigen anderen Distributionen übernommen wurde.

Diese Anleitung versucht nicht, apt-rpm (den Connectiva-Port von APT) zu behandeln, aber »Patches« für dieses Dokument, welche es tun, sind willkommen.

Diese Dokumentation basiert auf der nächsten Debian-Version: `Etch` (Stand: 31. August 2005).

## 2.1 Grundlegende Begriffe und Konzepte

Hier finden Sie einige grundlegende Begriffe und Konzepte, die in dieser Anleitung benutzt werden:

**APT-Quelle (englisch: APT source)**: eine APT-Quelle ist ein Ort (oft im Internet, möglicherweise eine CDROM oder ein anderer Ort), der als Repository von Debian-Paketen dient, siehe 'Die Datei /etc/apt/sources.list' auf Seite 7.

**APT-Quellenzeile (englisch: APT source line)**: Eine APT-Quellenzeile ist eine Zeile, die Sie einer Konfigurationsdatei hinzufügen, um APT die »APT-Quellen«, die Sie benutzen möchten, mitzuteilen, siehe 'Die Datei /etc/apt/sources.list' auf Seite 7.

**Binärpaket**: ein Binärpaket ist eine `.deb` Datei, die vorbereitet wurde, um vom Paketmanager (dpkg) installiert zu werden. Sie kann binäre Dateien, aber auch nur architektur-unabhändige Daten enthalten – es heißt in beiden Fällen Binärpaket.

**debian-nativ**: ein spezifisch für Debian erstelltes Paket, diese Art Pakete hat für gewöhnlich die Debian-Kontrolldateien innerhalb der Originalquellen, und jede neue Version des Pakets ist auch eine neue Version von Originalprogramm oder -daten.

**debianisieren**: ein Verb, das für gewöhnlich heißt: »Vorbereiten für die Benutzung mit Debian«, oder einfacher gesagt, paketieren im `.deb`-Format.

**Quellpaket (englisch: source package)**: »Quellpaket« ist in Wirklichkeit eine abstrakte Definition einer Menge von zwei oder drei Dateien, die Teil des Debian-Quellformates sind: eine `.dsc`-Datei, die Informationen über das Paket enthält, auch Quellkontrolldatei (englisch: source control file) genannt; eine `.orig.tar.gz`-Datei, welche die originalen Upstream-Quellen für dieses Paket enthält – Sie mögen auch Pakete finden, bei denen diese Datei einfach `.tar.gz` heißt, ohne ».orig«, was bedeutet, dass dies ein debian-natives Paket ist; und eine `.diff.gz`-Datei, die die Änderungen an den Originalquellen beim »Debianisieren« des Pakets enthält – bei debian-nativen Paketen finden Sie diese Art Datei nicht.

**Upstream**: dieses Wort meint für gewöhnlich etwas, das vom Original-Entwickler der Software oder Daten kommt, oder den Entwickler selbst.

**Virtuelle Pakete**: Dies sind Pakete, die nicht wirklich existieren, sondern generische Dienste sind, die von einigen spezifischen Pakete »bereitgestellt« werden – das bekannteste Beispiel ist das Paket `mail-transport-agent`, auf das Pakete, die einen MTA[1] benötigen, eine Abhängigkeit setzen können, während die Wahl des Benutzers, welchen MTA er benutzen will, frei bleibt.

[1]MTA bedeutet Mail Transfer Agent – es ist für gewöhnlich ein Server zum Verschicken und Empfangen von Email

# Kapitel 3

# Einführung in das Debian-Archiv

## 3.1 Einführung in Debian-Suiten

Die Debian-Entwicklung geschieht in einem Model, wo wir drei Haupt-»Zweige« haben, die wir »Suiten« nennen: Die stabile Suite (englisch: stable) ist die zuletzt zum Release gebrachte Version von Debian; die testing Suite enthält theoretisch eine Version, die stets fertig zum Release ist, Pakete kommen aus dem instabilen Zweig hierher, nachdem sie dort 10 Tage verbracht haben, keine kritischen Probleme berichtet wurden und sie auf allen Release-Architekturen gebaut wurden; in der instabilen Suite (englisch: unstable) passiert die Entwicklung: alle neuen Pakete kommen hierher, um darauf getestet zu werden, ob sie bereit zum Release sind, und gelangen dann nach testing.

Die stabile Suite erhält niemals neue Versionen von Paketen, nur neue Revisionen um Sicherheits- oder kritische Probleme zu beheben. Sie soll stabil bleiben, in dem Sinne dass der Administrator keine Veränderungerungen in Konfiguration oder Verhalten der Software, die er administriert, erwarten muss.

Es gibt eine weitere Suite, die von vielen Werkzeugen anders behandelt wird als die üblichen: die Suite experimental. Zu aller erst ist es keine in sich geschlossene Suite, d.h. Sie können nur aus ihr kein System installieren, für gewöhnlich müssen Sie sie zusammen mit einer der oben genannten benutzen: meistens wird die instabile gewählt.

Sie wird anders behandelt, weil sie in der Tat von höchst experimenteller Natur ist. Pakete, von denen ihre Betreuer denken, dass sie kaputt oder noch nicht bereit sind, in die instabile Suite zu gelangen, kommen in die experimentelle Suite, um ausgiebiger von furchlosen Betreuern getestet zu werden. Bitte benutzen Sie »experimental« nicht, wenn Sie nicht wirklich bei der Entwicklung helfen wollen, selbst achtgebend auf die Mühen, die es Ihnen bringen könnte.

## 3.2 Debians Sektionen

Jede Suite eines Debian-Repositories besteht für gewöhnlich aus einer oder mehreren Sektionen; Namen und Bestimmungen können von Distributor zu Distributor variieren. Debian

selbst hat drei Sektionen: main, contrib und non-free.

Die Sektion `main` ist das, was offiziell Debian wirklich ist. Alle Software, die konform ist zu den »Debian Free Software Guidelines« (http://www.debian.org/social_contract#guidelines) (Debians Richtlinien für Freie Software, kurz DFSG) und nur von DSFG-konformer Software abhängt, kommt in diese Sektion des Archivs, in Einklang mit dem »Social Contract« (http://www.debian.org/social_contract) (Debians Sozialvertrag). Durch diese Dokumente und das Versprechen, es nur Software, die mit ihnen im Einklang ist, zu erlauben, in die Sektion main zu gelangen, bietet Debian einen vernünftigen klaren Basisstandard Freiheit, auf den sich Benutzer verlassen können, wenn sie Software aus main installieren.

Die Sektion `contrib` wird von DFSG-konformer Software gebildet, die zum Arbeiten von nicht-DFSG-konformen Software oder Daten abhängt. Die Sektion `non-free` wird von nicht-DFSG-konformer Software gebildet, die aber verteilt werden darf. Sie sollten die Lizenz jeder Software aus non-free, die Sie installieren möchten, prüfen, um zu beurteilen, ob sie Ihnen gut genug ist, ihr zuzustimmen.

## 3.3 APTs Vertrauenskette

APT ist mit einer kryptographischen Vertrauenskette (englisch: chain of trust) ausgestattet, die sich von den Debian-Entwicklern bis zu den Endbenutzern erstreckt.

Die Kette beginnt bei den Paket-Betreuern, die ihre Pakete signieren und herauf laden. Jede Signatur wird gegen den Schlüssel des Paketieres geprüft, der herauf geladen wurde, als er ein offizieller Debian-Entwickler wurde. Dieser Schlüssel selbst wurde persönlich von anderen Debian-Entwicklern verifiziert.

Sobald verifiziert wurde, dass das herauf geladene Paket vom Betreuer signiert wurde, wird eine MD5-Prüfsumme des Pakets berechnet und in die Datei »Packages« geschrieben. Dann werden die MD5-Prüfsummen aller Packages-Dateien berechnet und in die Datei »Release« geschrieben. Dann wird diese mit dem Archiv-Schlüssel signiert, der einmal pro Jahr erzeugt und vom FTP-Server verteilt wird. Dieser Schlüssel ist auch im Debian-Schlüsselbund.

Daher kann der Endbenutzer, nachdem er den Archiv-Schlüssel seinem Schlüsselbund hinzugefügt hat, überprüfen, dass die Datei `Release` mit dem richtigen Schlüssel signiert wurde, die MD5-Prüfsummen aller `Packages`-Dateien und die MD5-Prüfsummen aller Debian-Pakete überprüfen. APT automatisiert diesen Prozess:

Während des Updates (siehe 'Update der Liste der verfügbaren Pakete' auf Seite 15) prüft APT die Signatur der Release-Dateien unter Benutzung seines Schlüsselbunds (für Informationen hierüber siehe 'Pakete für echt befinden' auf Seite 8) und die MD5-Prüfsummen der Packages-Dateien (die in den Release-Dateien aufgezeichet sind).

Beim Aktualisieren oder Installieren eines Paketes (siehe 'Aktualisieren von Paketen' auf Seite 21 bzw. 'Installieren und Reinstallieren von Paketen' auf Seite 16) prüft APT die MD5-Prüfsummen der Pakete (die in den Packages-Dateien gespeichert sind).

Wenn irgendeiner dieser Schritte scheitert, wird der Benutzer gewarnt und der Prozess unterbricht.

# Kapitel 4

# Basis-Konfiguration

## 4.1 Die Datei /etc/apt/sources.list

Als Teil seiner Arbeit benutzt APT eine Datei, die die »APT-Quellen«, von denen man Pakete beziehen kann, auflistet. Diese Datei heißt `/etc/apt/sources.list`.

Die Einträge in dieser Datei sind normalerweise von folgendem Format:

```
deb http://host/debian distribution sektion1 sektion2 sektion3
deb-src http://host/debian distribution sektion1 sektion2 sektion3
```

Natürlich sind obige Einträge erfunden und sollten nicht benutzt werden. Das erste Wort jeder Zeile, `deb` oder `deb-src` zeigt den Typ des Archivs: entweder es enthält Binär-Pakete (`deb`), das sind die vorkompilierten Pakete, die wir normalerweise benutzen, oder Quellpakete (`deb-src`), welche die originalen Programmquellen und die Debian-Kontrolldatei (`.dsc`) und das `diff.gz`, welches die Änderungen enthält, die für das »Debianisieren« des Programms von Nöten sind.

Wenn wir in dem obigen Beispiel das Wort `distribution` betrachten, definiert es, welche Debian Suite wir zum Ziel haben. Dies können generische Namen, wie »stable« und »testing«, oder spezifische, wie »sarge« und »etch«, sein. Sagen wir, das aktuelle testing nennt sich »etch«, und Sie wollen »etch« weiterverfolgen, selbst wenn es stabil wird; in diesem Falle sollten Sie »etch« für `distribution` verwenden. Wenn Sie dagegen stets testing verfolgen wollen, sollten Sie stattdessen »testing« verwenden.

Wenn Sie also immer die neuste stabile Distribution haben und, sobald ein neues Release herausgegeben wird, aktualisieren möchten, sollten Sie für `distribution` »stable« benutzen. Dies muss nicht unbedingt eine gute Idee sein, wenn Sie Aktualisierungen für stabile Veröffentlichungen planen möchten, sie beinhalten manchmal mehr als einfach ein paar Fragen zu beantworten, neben etwas Testen und Backuppen vor dem Fortfahren, deshalb empfehle ich Ihnen, stets die Kodenamen zu benutzen.

Normalerweise finden wir folgendes in der Standard-Debian-sources.list:

```
deb http://http.us.debian.org/debian stable main
deb http://security.debian.org stable/updates main

#deb-src http://http.us.debian.org/debian stable
#deb-src http://security.debian.org stable/updates main
```

Dieses sind die Zeilen, die eine Debian-Basis-Installation benötigt. Die erste `deb` Zeile zeigt auf das offizielle Archiv und die zweite auf das Archiv der Debian Sicherheits-Updates.

Die letzten beiden Zeilen sind auskommentiert (mit einem »#« am Anfang), deshalb ignoriert sie APT. Sie sind `deb-src`-Zeilen, das bedeutet, sie führen uns zu Debian-Quellpaketen. Wenn Sie öfters Programm-Quellen herunterladen, um sie zu testen oder neu zu kompilieren, sollten Sie die Kommentarzeichen entfernen.

Die Datei `/etc/apt/sources.list` kann verschiedene Typen von Zeilen enthalten. APT kann mit Archiven der Typen `http`, `ftp`, `file` (lokale Dateien, z.B. ein Verzeichnis, mit einem gemounteten ISO9660-Dateisystem) und `ssh`, soweit ich weiß.

Merken Sie: jedesmal wenn Sie eine Quelle, von der Pakete bezogen werden sollen, für APT hinzufügen möchten, ist dieses die Datei die Sie bearbeiten.

Vergessen Sie nicht, die Liste von Paketen zu aktualisieren (siehe 'Update der Liste der verfügbaren Pakete' auf Seite 15), nachdem die Datei `/etc/apt-/sources.list` editiert wurde. Dies ist notwendig, damit APT weiß, welche Pakete von den spezifizierten Quellen verfügbar sind.

## 4.2 Pakete für echt befinden

Seit APT-Version 0.6 werden Pakete auf ihre Echtheit geprüft, um sicherzustellen, dass sie von der Quelle ausgehen, von der sie vorgeben zu kommen. Dies ist ein zusätzliches Sicherheitsfeature. Wenn das System die bei der Installation eines Paketes die Echtheit nicht feststellen kann, werden Sie gefragt, ob Sie die Installation abrechen wollen.

Da diese Echtheitsprüfung auf kryptographischen Methoden basiert, verwaltet APT seinen eigenen Schlüsselbund. Jeder der Einträge in der Datei `sources.list` hat einen korrespondierenden Schlüssel. Wenn Sie jedoch inoffizielle APT-Quellen benutzen, ist es möglich, dass eine Quelle, die Sie benutzen, nicht auf Echtheit überprüft wird. Wenn dies der Fall ist, sollten Sie den Betreuer dieser Quelle ermuntern, Echtheitsprüfung für sein Angebot zu implementieren.

Um aus diesem Sicherheitsfeature Gewinn zu ziehen, müssen Sie für jede auf auf Echtheit zu prüfende Quelle einen Schlüssel zu APTs Schlüsselbund hinzufügen. Dies kann mit GPG erledigt werden, aber APT bietet ein Werkzeug, `apt-key`, das eine vereinfachte Schnittstelle für GPG ist.

`apt-key` ist einfach zu benutzen. Der schwierige Teil des Prozesses ist es, einen Schlüssel für jede der Quellen zu besorgen, und sicherzustellen, dass Sie diesem Schlüssel vertrauen können.

Debians Archivschlüssel wird in `/usr/share/apt/debian-archive.gpg` installiert, so dass Sie einfach diese Datei benutzen können, um den offiziellen Archivschlüssel zu Ihrem APT-Schlüsselbund hinzuzufügen, indem Sie dies eingeben:

```
# apt-key add /usr/share/apt/debian-archive.gpg
```

Für unoffizielle Quellen außerhalb Debian müssen Sie herausfinden, wo sie ihren öffentlichen Schlüssel bereitstellen, so dass Sie ihn in Ihren APT-Schlüsselbund importieren können.

Wenn Sie es aus einem bestimmten Grund vorziehen, die kryptographische Überprüfung der Release-Dateien zu deaktivieren, können Sie folgendes zu APTs Konfiguration hinzufügen (siehe 'Die APT Konfigurationsdatei' auf der nächsten Seite):

```
APT::Get::AllowUnauthenticated "true";
```

## 4.3 Hinzufügen einer CD-ROM in die Datei sources.list

Wenn Sie lieber eine CD-ROM zum Installieren von Paketen oder automatischen Updaten des Systems durch APT verwenden möchten, können Sie sie in Ihre `sources.list` eintragen. Um dieses zu tun, können Sie das Programm `apt-cdrom` benutzen, wie im folgenden beschrieben:

```
# apt-cdrom add
```

mit der Debian-CD-ROM im Laufwerk. Die CD-ROM wird gemountet, und wenn sie eine gültige Debian-CD ist, wird auf ihr nach Paketinformationen gesucht. Wenn Ihre CD-ROM-Konfiguration ein wenig ungewöhnlich ist, können Sie auch die folgenden Optionen benutzen:

```
-h             - Programmhilfe
-d Verzeichnis - Einhängepunkt der CD-ROM
-r             - Eine erkannte CD-ROM umbenennen
-m             - kein Einhängen
-f             - Schneller Modus, überprüft keine Paket Dateien
-a             - Gründlicher Scan-Modus
```

Zum Beispiel:

```
# apt-cdrom -d /home/kov/mycdrom add
```

Sie können eine CD-ROM auch identifizieren, ohne sie zur sources.list hinzuzufügen:

```
# apt-cdrom ident
```

Beachten Sie, dass dieses Programm nur funktioniert, wenn Ihr CD-ROM Laufwerk in der `/etc/fstab` ihres Systems korrekt konfiguriert ist.

## 4.4 Die APT Konfigurationsdatei

APT benutzt /etc/apt/apt.conf als seine Hauptkonfigurationsdatei. Obwohl es keine so benannte Datei in einer Standardinstallation gibt, wie Sie herausfinden werden, können Sie sicher eine erzeugen und editieren. Es gibt einen modularen Weg, die Konfiguration zu bewerkstelligen, wenn Sie dies vorziehen: Sie können einzelne Dateien mit beliebigen Namen in /etc/apt/apt.conf.d/ platzieren.

Hüten Sie sich vor zwei Tatsachen, wenn Sie den modularen Weg wählen: einige Debian-Pakete legen ihre eigene Konfiguration in dieses Verzeichnis, so dass Sie versuchen müssen, Namenskollisionen zu vermeiden, zum Beispiel indem Sie ein Suffix -local an den Namen anhängen. Desweiteren wird die Konfiguration geordnet nach Dateinamen gelesen, so dass Sie eine Nummer ganz am Anfang des Dateinamens hinzufügen können, um sie in der Folge zu positionieren. Zum Beispiel können Sie sie 00000myconf-local nennen, wenn Sie sicherstellen wollen, dass sie die erste berücksichtigte Konfigurationsdatei ist.

Um die Syntax und Optionen kennen zu lernen, die von diesen Konfigurationsdateien akzeptiert werden, siehe die Manpage apt.conf(5).

## 4.5 APT-Prioritäten für Pakete anpassen

APT benutzt einen Priorisierungsalgorithmus, um zu entscheiden, von welchem Repository es ein gegebenes Paket holen soll. Hier ist ein einfaches Beispiel:

```
$ apt-cache policy apt-howto
apt-howto:
  Installiert: 1.8.10.3-1
  Mögliche Pakete: 1.8.11-1
  Versions-Tabelle:
     1.8.11-1 0
        500 http://ftp.nl.debian.org sid/main Packages
 *** 1.8.10.3-1 0
        500 http://ftp.nl.debian.org sarge/main Packages
        100 /var/lib/dpkg/status
```

Ich habe hier zwei Quellen: eine für »sid« und eine für »sarge«, und das ist alles.

Die installierte Version wird mit *** markiert. Wir können sehen, dass sie installiert ist, weil auch die Datei /var/lib/dpkg/status erwähnt wird, die die Informationen über den aktuelle Zustand des Systems enthält. Wir können sehen, dass das Paket aus »sarge« kommt. Die Version aus »sid« ist ebenfalls als verfügbar gelistet.

APT gibt installierten Paketen Priorität 100, wie wir sehen können, und 500 für alle anderen Quellen, mit einer einzigen Ausnahme: die Suite experimental, die, wie wir bereits sagten, von vielen Werkzeugen auf spezielle Art behandelt wird. APTs Algorithmus zieht es vor, Pakete von Quellen mit höherer Priorität zu installieren. Wenn die Prioritäten die selben sind, zieht

es die höchste Version vor. Sie können dies beim Betrachten des Feldes `Mögliche Version` sehen, welches die neuere Version, aus »unstable«, als mögliche Version zur Installation listet: APT will sie aktualisieren.

Wenn Sie nur ausgewählte Pakete aus der Quelle »sid« installieren möchten, und nicht möchten, dass APT automatisch Pakete von dieser Quelle als mögliche Versionen für Aktualisierungen betrachtet, dann müssen Sie an seinen Prioritäten drehen. Sie tun dies, indem Sie das Feld `Archive` der `Release`-Datei der Quelle, der Sie Priorität geben möchten, benutzen. Sie können diese Information herausfinden, indem Sie in die Release-Datei schauen, die der Prozess des Updatens heruntergeladen hat:

```
# grep ^Archive /var/lib/apt/lists/ftp.nl.debian.org_debian_dists_sarge_main_
Archive: stable
```

Beachten Sie, dass sich der Dateiname in Abhängigkeit von Ihrer Quellenzeile ändert. Um nun APT dazu zu bringen, die Pakete auf der stabilen Suite zu halten, fügen Sie eine Datei folgenden Inhaltes zu `/etc/apt/apt.conf.d/` hinzu:

```
APT::Default-Release "stable";
```

Danach hat sich APTs Strategie (englisch: policy) geändert:

```
$ apt-cache policy apt-howto
apt-howto:
  Installiert: 1.8.10.3-1
  Mögliche Version: 1.8.10.3-1
  Versions-Tabelle:
     1.8.11-1 0
        500 http://ftp.nl.debian.org sid/main Packages
 *** 1.8.10.3-1 0
        990 http://ftp.nl.debian.org sarge/main Packages
        100 /var/lib/dpkg/status
```

APT gibt seiner Standard-Quelle Priorität 990 für jedes Paket, die anderen Quellen bleiben genau wie vorher. Pakete aus anderen Quellen, die installiert sind, haben ihre Priorität von 100 auf 500 erhöht. Warum wird sie APT nicht deaktualisieren? Weil APT eine Deaktualisierung nur in Betracht zieht, wenn etwas eine Priorität über 1000 erhält. Das heißt aber, dass APT nicht automatisch die Versionen von den Nicht-Standard-Quellen aktualisiert, solange das Werkzeug, das Sie benutzen, Ihnen dabei helfen will, was der Fall ist bei `aptitude`.

Prioritäten für Gruppen von Paketen, basierend auf vielen anderen Kriterien, definieren ist ebenfalls möglich, durch Nutzen der Konfigurationsdatei `/etc/apt/preferences`. Sie können APT sogar überzeugen, Ihr ganzes System zu deaktualisieren, obwohl das nicht unterstützt wird, und nicht ausprobiert werden sollte, wenn Sie ein schwaches Herz haben. Sie können hierüber mehr Informationen bekommen, indem Sie die Manpage `apt_preferences(5)` lesen.

## 4.6 APT durch einen Proxy hindurch benutzen

Wenn Sie APT in einem Netzwerk benutzen, in dem alle HTTP- und FTP-Verbindungen durch einen Proxy laufen, müssen Sie APT so aufsetzen, dass es diesen Proxy benutzt. Sie können dies tun, indem Sie die Konfigurationdatei `/etc/apt/apt.conf` editieren oder eine Konfigurationsdatei in dem Verzeichnis `/etc/apt/apt.conf.d/` platzieren, was es einfacher macht, den Konfigurationskram zu organisieren (siehe 'Die APT Konfigurationsdatei' auf Seite 10).

Wenn diese Datei nicht existiert, dann erzeugen Sie sie und fügen Zeilen wie die folgenden hinzu:

```
Acquire::http::proxy "http://Proxy:Port";
Acquire::ftp::proxy "http://Proxy:Port";
```

Ersetzen Sie *Proxy* und *Port* durch das, was Ihnen Ihr Netzwerk-Administrator gegeben hat. Sie können auch einen Benutzer und ein Passwort angeben, wenn es der Proxy erfordert, so wie hier:

```
Acquire::http::proxy "http://Benutzer:Passwort@Proxy:Port";
```

Es gibt viele andere nützliche Optionen, die Sie für APT in der Konfigurationsdatei `/etc/apt/apt.conf` setzen können, siehe 'Die APT Konfigurationsdatei' auf Seite 10.

# Kapitel 5

# Paketverwaltung

## 5.1 Debians Paketmanager benutzen: »aptitude«

Debian hat lange mit einem Paketmanager gelebt, der bekannt dafür war, schwierig zu verstehen und zu benutzen zu sein, genannt `dselect`. Das Debian-System bewegt sich zu einem neuen Standard-Manager, genannt `aptitude`, der ebenfalls auf der APT-Bibliothek basiert.

Wenn Sie »Sarge« installiert haben, haben Sie `aptitude` im Basissystem installiert, ansonsten installieren Sie, bevor Sie fortfahren, das Paket `aptitude`, zum Beispiel, indem Sie dieses Kommando ausführen:

```
# apt-get install aptitude
```

Versuchen Sie, die Benutzung von `apt-get` zu vermeiden und stattdessen `aptitude` zu benutzen, weil es viel mehr über die Fähigkeiten von APT weiß und einige einzigartige Features hat, wie das Markieren von Paketen, die automatisch installiert wurden, um Abhängigkeiten zu erfüllen, und das Fragen, diese automatisch zu entfernen, wenn sie nicht mehr gebraucht werden.

`aptitude` hat zwei Hauptmodi: eine ncurses-basierte Benutzerschnittstelle, in der Sie durch ein Liste von Paketen navigieren können, fast wie in `dselect`, und eine Befehlszeilen-basierte Benutzerschnittstelle, fast wie `apt-get`. Die meisten Beispiele in diesem Dokument benutzen die Befehlszeilen-basierte Schnittstelle, Tipps gebend, welches Kommando benutzt wird, um dieselben Resultate in `aptitude`s dselect-ähnlicher Schnittstelle zu erreichen.

Um diese Schnittstelle zu starten, geben Sie in einem Terminal `aptitude` ein – Sie müssen zu diesem Zeitpunkt nicht »root« sein, aptitude fordert automatisch Rootrechte an, wenn es sie braucht. Sehen wir uns die Grundlagen an: um die Menüzeile oben auf dem Bildschirm zu erreichen, müssen Sie »`F10`« drücken, wie die zweite Zeile vorschlägt. Diese zweite Zeile hat eine Zusammenfassung der meistgenutzten Features, wie Hilfe bekommen, Beenden, die Liste der Pakete aktualisieren und Fortfahren.

Werfen Sie einen Blick in das `Handbuch`, welche sich im Menü `Hilfe` befindet, um zu verstehen, wie das Ding arbeitet.

Beachten Sie, dass `aptitude` auf eine Weise arbeitet, die ähnlich anderen Paketmanagern ist: Sie machen alle Ihre Auswahlen und `kommitieren` dann die Änderungen. Fast alle zu erledigenden Aktionen beim Managen der Installation eines Paketes erfordern es, die Taste »`g`« zwei Mal zu drücken, um in Aktion zu treten. Das erste Mal, wenn Sie »`g`« drücken, wird Ihnen die Liste der zu erledigenden Änderungen präsentiert, dann drücken Sie nochmals »`g`«, um die Änderungen zu bestätigen, oder »`q`« um zurück zu gehen,

Ähnlich können Sie die »`Eingabe`«-Taste drücken, um mehr Details über ein Paket oder eine Gruppe von Paketen zu bekommen – zum Beispiel können Sie »`Eingabe`« drücken, wenn der Auswahlbalken auf »Neue Pakete« ist, um eine Liste von Sektionen zu sehen. Durch Drücken von »`Eingabe`« auf einer Sektion sehen Sie die Liste neuer Pakete in dieser Sektion, und Sie können mehr Details zu einem Paket sehen, indem Sie auf ihm »`Eingabe`« drücken. Beachten Sie, dass das Betreten der Detailansicht eines Paketes `aptitude`s Sichtkontext ändert, so dass Sie »`q` drücken müssen um zurück zu gelangen.

Die Pakete sind standardmäßig basierend auf ihrem Installationsstatus und einigen anderen besonderen Kriterien kategorisiert. Sie sehen folgendes, wenn Sie das erste Mal `aptitude` betreten:

```
--- Neue Pakete
--- Aktualisierbare Pakete
--- Installierte Pakete
--- Nicht installierte Pakete
--- Veraltete und selbst erstellte Pakete
--- Virtuelle Pakete
--- Schnellauswahl (Tasks)
```

»Neue Pakete« hat die Pakete, die zu Debian dazugekommen sind, seit Sie Ihre Liste neuer Pakete, durch Benutzen der Option `Neue Pakete vergessen` im Menü `Aktionen` oder der Taste »`f`«, geleert haben.

»Aktualisierbare Pakete« sind installierte Paket, die neue Versionen verfügbar haben. »Installierte Pakete« sind die korrekt installierten Pakete. »Nicht installierte Pakete« sind die, die Sie nicht installiert haben, die aber von einer der APT-Quellen verfügbar sind. »Veraltete und selbst erstellte Pakete« sind diejenigen Pakete, die nicht mehr von mindestens einer APT-Quelle verfügbar sind, oder solche, die Sie selbst erzeugt haben. »Virtuelle Pakete« sind Pakete, die einen Dienst statt einer bestimmten Software definieren, und die von mehreren spezifischen Paketen »bereitgestellt« werden. Schließlich sind »Schnellauswahl (Tasks)« eine Sammlung von Paketen, die einen bestimmten Zweck dienen, sie werden auch von dem Programm `tasksel` benutzt.

»`Enter`« auf einem von diesen drücken erweitert den Baum ungefähr so:

```
--\ Nicht installierte Pakete
  --- admin - Administrator-Werkzeuge
  --- base - Das Debian-Grundsystem
[...]
```

Noch tiefergehend haben wir:

```
  --\ games - Spiel & Spaß
    --- contrib - Programme, die von Nicht-Debian-Software abhängen
    --\ main - Die Debian-Distribution
p      3dchess                                          <keine>    0.8.1-1
```

Dies zeigt uns ein Paket, das nicht installiert ist, das ein Spiel ist und das vom Debian-Repository »main« verfügbar ist, was das ist, was die Distribution Debian offiziell ist.

In der Mitte des Bildschirms ist eine markierte Zeile, die sich mit etwas Text füllt, wenn Sie ein Paket auswählen. Das nennen wir `Kurzbeschreibung` (englisch: `short description`) des Paketes. Darunter ist ein Kasten, der detailiertere Informationen über das Paket enthält. Das ist die `lange Beschreibung` (englisch: `long description`. Manches Mal ist die lange Beschreibung zu groß, um in das für sie reservierte Rechteck zu passen, dann können sie die Taste »`z`« zum Herunter-Scrollen drücken und »`a`« um wieder hoch zu scrollen.

Für mehr Informationen über die hier gezeigten Daten siehe ‘Installieren und Reinstallieren von Paketen’ auf der nächsten Seite.

## 5.2 Update der Liste der verfügbaren Pakete

Das Paketsystem benutzt eine private Datenbank um Buch zu führen über installierte, nicht installierte und für eine Installation verfügbare Pakete. Das Programm `aptitude` benutzt diese Datenbank, um herauszufinden, wie es die vom Benutzer angeforderten Pakete installieren soll und welche zusätzlichen Pakete benötigt werden, damit die ausgewählten Pakete ordentlich funktionieren.

Um diese Liste zu updaten, benutzen Sie das Kommando `aptitude update`. Dieses Kommando sucht nach den Paketlisten in den Archiven aus der `/etc/apt/sources.list`; siehe ‘Die Datei /etc/apt/sources.list’ auf Seite 7 für weitere Information über diese Datei. Sie können auch die Taste »`u`« in der Ncurses-Schnittstelle von `aptitude` benutzen, um es die Paketlisten updaten zu lassen.

Einer der Schritte die APT beim Updaten Ihrer Datenbank unternimmt, ist es zu verifizieren, dass die Liste von Paketen und ihre MD5-Prüfsummen sicher ist. Es tut dies durch Verifizieren einer kryptographischen Signatur und Überprüfen einiger MD5-Prüfsummen, für mehr Informationen siehe ‘APTs Vertrauenskette’ auf Seite 6. Wenn APT findet, dass die Signatur inkorrekt ist oder dass die MD5-Prüfsummen nicht passen, gibt es Ihnen eine Fehlermeldung aus während des Updatens Ihrer Paketdatenbank. Dies geschieht, um zu vehindern, dass trojanisierte Pakete auf Ihrem Computer installiert werden. Wenn dies auftritt, könnte es ein Konfigurationsproblem geben (etwa ein abgelaufener Schlüssel), das Sie beheben können. Scheitert dies, sollten Sie den Administrator der scheiternden APT-Quelle kontaktieren.

Es ist eine gute Idee, die Listen der verfügbaren Pakete regelmäßig zu updaten, um sich selbst und sein System auf dem neusten Stand über mögliche Paket- bzw. Sicherheitsupdates zu halten.

## 5.3 Nach Paketen suchen

aptitude bietet eine sehr flexible und mächtige Suchfunktion. Sie können auf sie zugreifen, indem Sie die Taste »/« (Schrägstrich) drücken. Dies öffnet einen kleinen Dialog in der Mitte des Bildschirms.

Dann können Sie den Namen eines Paketes tippen. aptitude sucht, während Sie tippen. Wenn das Paket, das es findet, nicht genau das ist, was Sie wollen, können Sie »Eingabe« drücken, dass der Dialog verschwindet, und dann die Taste »n«, um die Suche nach derselben Zeichenkette zu wiederholen.

Sie können im Suchdialog reguläre Ausdrücke[1] benutzen, hüten Sie sich nur davor, dass aptitude seine eigene Such-Sprache hat, die sich manchmal mit der Sprache der regulären Ausdrücke überlappt. Gehen wir nun zu einem Beispiel: Wenn Sie einen exakten Treffer auf die Zeichenkette »gnome« haben wollen, geben Sie »^gnome$« im Suchdialog ein.

Beachten Sie, dass, wenn Sie so suchen, Sie nach einfachen Paketnamen suchen. Wenn Sie in den Beschreibungen suchen wollen, müssen Sie »~d« direkt vor dem Suchausdruck benutzen. Falls Sie also nach allen Paketen suchen wollen, die »isolinux« in ihrer Beschreibung erwähnen, werden Sie »~disolinux« im Suchdialog schreiben.

Sie können auch nach Paketen suchen, die gegenwärtig kaputt sind, indem Sie nach »~b« suchen, oder mit »~c« nach Paketen suchen, die entfernt wurden, deren Konfigurationsdateien aber erhalten wurden.

Für weiter fortgeschrittenes Suchen in aptitude werfen Sie einen Blick in sein Handbuch, das Sie im Menü Hilfe finden können.

Sie können auch die Befehlszeile benutzen, um nach Paketen zu suchen, wie hier:

```
$ aptitude search '~c'
c A abiword-common                    - WYSIWYG word processor based on GTK2
c A acme                              - Enables the "multimedia buttons" found
(...)
```

Wie Sie sehen können, können Sie jede von aptitudes Suchzeichenketten auch auf der Befehlszeile benutzen, und aptitude bietet Ihnen einige Informationen zum Zustand von Paketen, die wir später in diesem Dokument diskutieren werden.

## 5.4 Installieren und Reinstallieren von Paketen

Endlich kommt das, worauf Sie alle gewartet haben! Mit Ihrer fertigen sources.list, den Schlüsseln zu APTs Schlüsselbund hinzugefügt, und Ihrer Liste der verfügbaren Pakete auf dem

[1] Reguläre Ausdrücke sind besondere Zeichenketten, um Übereinstimmungen basierend auf bestimmten Bedigungungen zu finden, werfen Sie einen Blick auf die Manpage regex(7)

neusten Stand ist alles, was Sie zu tun haben, `aptitude` auszuführen, um Ihr gewünschtes Paket zu installieren.

Wenn Sie `aptitude` betreten, können Sie es ein oder mehrere Pakete installieren lassen, indem Sie die Taste »+« drücken, während das fragliche Paket ausgewählt ist. Sie können die Taste »/« benutzen, um nach dem Paket, das Sie wollen, zu suchen, siehe ‘Nach Paketen suchen’ auf der vorherigen Seite. Wenn Sie die Installation eines Paketes anfordern, wird es grün markiert, und die Zeile sieht aus wie in diesem Beispiel:

```
pi    celestia-gnome                                          <keine>    1.3.0-
```

Dies bedeutet, dass das Paket nicht installiert ist (»p«) und zur Installation markiert ist (»i«). Das Wort `<keine>` bedeutet, keine Version ist installiert, und die Versionsnummer danach ist die, die zur Installation markiert wurde.

Beachten Sie, dass wenn Sie dies tun, andere Pakete automatisch zur Installation markiert werden. Dies sind die Abhängigkeiten, Vorschläge und Empfehlungen des ausgewählten Paketes. Sie werdem höchstwahrscheinlich so markiert:

```
piA   gnome-bin                                               <keine>    1.4.2-
```

Das große »A« bedeutet, dass dies Paket installiert wird, weil ein anderes Paket von ihm abhängt. Alle so installierten Paket werden entfernt, wenn sie nicht mehr gebraucht werden. Sie können diese Markierung von Hand setzen, indem sie »`M`« tippen, während der Cursor auf einem Paket positioniert ist.

Ein »U« in der vierten Spalte bedeutet, dass das Paket von einer Quelle kommt, der nicht vertraut wird, was bedeutet, dass es keine signierte `Release`-Datei für die Quelle gibt, von der das Paket kommt, oder dass ihr öffentlicher Schlüssel nicht in APTs Schlüsselbund importiert wurde. Siehe ‘Pakete für echt befinden’ auf Seite 8 um zu verstehen, was Sie tun sollten, dass dieser Quelle vertraut wird. Sie könnten dasselbe Paket allerdings auch in einer der Quellen, denen vertraut wird, haben, so dass Sie einen Blick auf die detailierten Informationen über verfügbare Versionen dieses Paketes werfen möchten.

Sie können detailierte Informationen über das Paket, das Sie installieren wollen, bekommen, indem Sie »`Eingabe`« drücken, während das Paket ausgewählt ist. Dies bringt Sie zu etwas wie diesem hier:

```
i A --\ xterm                                              4.2.1-14     4.2.1-
  Beschreibung: xterm - X terminal emulator
    xterm is a terminal emulator for the X Window System.  It provides DEC
    VT102 and Tektronix 4014 compatible terminals for programs that cannot
    use the window system directly.  This version implements ISO/ANSI
    colors and most of the control sequences used by DEC VT220 terminals.
  Priorität: optional
  Bereich: x11
```

```
    Verwalter: Branden Robinson <branden@debian.org>
    Komprimierte Größe: 547k
    Unkomprimierte Größe: 1130k
    Quellpaket: xfree86
    --\ Hängt ab
      --- libc6 (>= 2.3.2.ds1-4)
      --- libfreetype6 (>= 2.1.5-1)
      --- libncurses5 (>= 5.3.20030510-1)
      --- libxaw7 (> 4.1.0)
      --- xlibs (> 4.1.0)
      --- xlibs (> 4.2.0)
    --\ Kollidiert
      --- xbase (< 3.3.2.3a-2)
      --- suidmanager (< 0.50)
    --\ Ersetzt
      --- xbase (< 3.3.2.3a-2)
    --- Pakete, die von xterm abhängen
    --\ Versionen
  p A 4.3.0-0pre1v4
  i A 4.2.1-14
```

Die Versionen, denen vertraut wird, haben nicht den Buchstaben »U« in der vierten Spalte, so dass Sie einfach die, die man haben möchte, herauspicken und »aptitude« installieren lassen können. Unser obiges Beispiel hat verschiedene Versionen, die von Quellen kommen, denen vertraut wird.

Beachten Sie, dass fast immer, wenn Sie drei oder mehr Bindestriche sehen, Sie »`Eingabe`« drücken können, um sie aufklappen zu lassen. Sie werden dann zu `--\`. Sie haben nun viele Informationen über dies Paket, etwa von welchen Paketen es abhängt und mit welchen es kollidiert[2] zum Beispiel.

Beachten Sie auch, dass Sie am Ende des Bildschirms (sie können mit den »`Pfeiltasten`« oder »`Bild ab`« nach unten gehen) die verschiedenen Versionen finden, die für das Paket verfügbar sind. Das Paket `xterm` hat für mich zwei verfügbare Versionen, weil ich mehr als eine APT-Quelle habe, die es in verschiedenen Versionen anbieten.

Sie können die Version, die Sie wollen, installieren, indem Sie auf ihr »+« drücken. Erinnern Sie sich, dass sie `aptitude` anweisen müssen, mit den Markierungen, die Sie gemacht haben, fortzufahren. Sie tun dies, indem Sie »`g`« drücken. `aptitude` zeigt Ihnen dann eine Liste mit den Aktionen, die es ausführen will, wo Sie sehen können, ob es tut, was Sie wollen:

```
  --\ Pakete, die automatisch installiert werden, um Abhängigkeiten zu erfüllen
  ciA calctool                                               <keine>    4.3.16
  --\ Pakete, die zurückgehalten werden
```

[2] Ein Paket, das mit einem anderen oder mit einer spezifischen Version eines anderen Paketes kollidiert, kann nicht zur gleichen Zeit installiert sein, wie dies Paket oder diese Paketversion.

```
ih  libgnomeprintui2.2-0                              2.4.0-1     2.4.2-
--\ Pakete, die installiert werden
pi  gwget2                                            <keine>     0.7-3
--\ Pakete, die entfernt werden
ip  qvwm                                            1:1.1.12-1    1:1.1.
```

Die Einträge sollten leicht verständlich sein. Das Neue hier ist, dass ein Paket zurückgehalten wird. Sie können darüber mehr unter 'Pakete auf »halten« setzen' auf Seite 22 lesen. Nachdem Sie geprüft haben, dass die Änderungen die gewünschten sind, bestätigen Sie die Auswahl durch nochmaliges Drücken der Taste »g«.

aptitude startet nun den Prozess des Herunterladens und Installierens der Pakete, die Sie ausgewählt haben. APT mag auch das Aktualisieren von Paketen beginnen, die zu dieser Zeit aktualisiert werden müssen (siehe 'Aktualisieren von Paketen' auf Seite 21). Wenn Sie Pakete von Quellen, denen nicht vertraut wird, ausgewählt haben, zeigt Ihnen aptitude eine Liste dieser Pakete und erwartet eine Eingabe von Ihnen, bevor es beginnt die Pakete zu holen, ob Sie sie wirklich installieren möchten. Siehe 'APTs Vertrauenskette' auf Seite 6 für mehr Informationen über sichere Quellen.

Wenn Sie ein Paket reinstallieren wollen, zum Beispiel um Dateien, die Sie versehentlich entfernt haben, zurückzuholen, können Sie die Taste »L« drücken, um es zur Reinstallation zu markieren. Sie gehen dann durch denselben Prozess, als wenn Sie es Installieren würden.

Sie können aptitude auch ohne die interaktive Benutzerschnittstelle benutzen, um Pakete zu installieren oder reinstallieren:

```
# aptitude install Paket1 Paket2 ...
# aptitude --reinstall install Paket1 Paket2 ...
```

## 5.5 Manuell heruntergeladene oder lokal erzeugte Debian-Pakete installieren

Wenn Sie eine .deb-Datei von Hand aus dem Internet herunter geladen, oder Sie ein .deb von einer Quelle haben, die APT nicht benutzt, können Sie aptitude oder irgendeinanderes APT-basiertes Programm nicht benutzen, um es zu installieren.

Sie müssen dann das darunterliegende Paket-Management-Werkzeug benutzen, welches tatsächlich die harte Arbeit des Paketeinstallierens erledigt – sogar aptitude ruft dieses Werkzeug zum Installieren auf: dpkg.

Sie können folgendermaßen ein Paket installieren:

```
# dpkg -i Datei.deb
```

Wenn Sie den Zweig unstable von Debian benutzen oder Debian-Pakete erzeugen möchten, sollten Sie besser mehr über dpkg heraus finden! Es ist ein sehr mächtiges Werkzeug und hat viele nützliche Optionen.

## 5.6 Pakete entfernen

Falls Sie ein Paket nicht mehr benutzen möchten, können Sie es unter Benutzung von APT von Ihrem System entfernen. Um dies zu tun, tippen Sie einfach »`-`«, während Sie das Paket in `aptitude` ausgewählt haben. Das markiert das Paket zum Entfernen mit einer purpurfarbenen Zeile, wie hier:

```
id    celestia-gnome                                   1.3.0-1     1.3.0-
```

Beachten Sie, dass dies mit einer Zeile endet, die wie folgt aussieht, nachdem das Paket entfernt ist:

```
c    ocaml-base                                         <keine>    3.07.2
```

Da ist ein ein »`c`« am Anfang der Zeile, statt eines »`p`«s. Das heißt, dass auf meinem System das Paket `ocaml-base` entfernt wurde, aber die Konfigurationsdateien immer noch herum liegen. Um das Paket mitsamt seinen Konfigurationsdateien zu entfernen, müssen Sie stattdessen die Taste »_«[3] benutzen – dies heißt vollständig entfernen oder, auf englisch, purge.

Eine wichtige Anmerkung: Falls Sie versuchen, ein Paket zu entfernen, das eine Abhängigkeit eines installierten Paketes ist, enden Sie mit kaputten Paketen, wie oben im `aptitude`-Bildschirm gezeigt:

```
aptitude 0.2.13      #Kaputt: 1   Werde 208MB freigeben
```

Die kaputten Pakete werden wie folgt markiert:

```
iBA   openoffice.org                                   1.1.0-3     1.1.0-
```

Sie können versuchen, die Kaputtheit selbst zu lösen, indem Sie die fehlerhaften Pakete ebenfalls entfernen lassen, so dass keine verletzten Abhängigkeiten zurückbleiben, oder `aptitude` durch Drücken der Taste »`g`« bitten fortzufahren. Es versucht dann, die ganze Kaputtheit zu reparieren. Wenn die von dem Programm gewählte Lösung nicht Ihren Wünschen entspricht, können Sie es von Hand reparieren, bevor Sie mit »`g`« bestätigen.

Sie können nach kaputten Paketen suchen, indem Sie »`/`« tippen, »`~b`« als Suchzeichenkette benutzen und »`Eingabe`« drücken; Sie können dann mit der Taste »`\`« nach dem nächsten Treffer suchen. Sie können kaputte Pakete auch suchen, indem Sie einfach die Taste »`b`« drücken; nochmaliges Drücken zeigt Ihnen das nächste kaputte Paket. Siehe 'Informationen über Pakete bekommen' auf Seite 33 für mehr Informationen wie man Pakete findet.

Wie üblich können Sie auch Pakete durch die Befehlszeilenschnittstelle entfernen oder vollständig entfernen lassen:

```
# aptitude remove Paket1 Paket2 ...
# aptitude purge Paket1 Paket2 ...
```

[3] Die Unterstrich-Taste, gewöhnlich zusammen mit dem Bindestrich zu finden, zu tippen durch Halten der Umschalttaste und Drücken der Bindestrich-Taste.

## 5.7 Aktualisieren von Paketen

Paket-Aktualisierungen (engl: upgrades) sind ein großer Erfolg des APT-Systems. `aptitude` markiert automatisch die Pakete, die neuere Versionen haben, zur Aktualisierung, so dass alles, was Sie für gewöhnlich tun müssen, um Ihr System zu aktualisieren, das Updaten der Paketlisten (siehe 'Update der Liste der verfügbaren Pakete' auf Seite 15) und dann `aptitude` durch Tippen von »`g`« bitten fortzufahren und nach Besichtigung der Veränderungen wieder mit »`g`« bestätigen ist. Falls `aptitude` die Aktualisierungen nicht automatisch markiert, können Sie die Taste »`U`« (großes »U«) drücken, um es dies tun zu lassen. Dies ist eine Konfigurations-Option, die sich im Menü `Optionen->Verschiedenes` befindet.

Sie können die Liste der Pakete, die neue Versionen verfügbar haben, überprüfen, indem Sie in die Kategorie `Aktualisierbare Pakete` schauen:

```
--\ Aktualisierbare Pakete
  --\ libs - Bibliotheken
    --\ main - Die Debian-Distribution
iu    libgnomeprintui2.2-0                               2.4.0-1     2.4.2-
```

Zur Aktualisierung markierte Pakete haben eine zyan-farbene Zeile und den Buchstaben `u` als die auszuführende Aktion. Falls Sie auf eine neue Veröffentlichung aktualisieren möchten, schauen Sie in den nächsten Abschnitt ('Aktualisierung auf eine neue Debian-Version' auf dieser Seite).

## 5.8 Aktualisierung auf eine neue Debian-Version

APT erlaubt es Ihnen, ein ganzes Debian-System auf einmal zu aktualisieren, entweder über das Internet oder von einer neuen CD-ROM (gekauft oder als ISO-Abbild aus dem Netz heruntergeladen).

Wenn auf Ihrem System z.B. Revision 0 der stabilen Debian-Version läuft und Sie sich Revision 3 auf CD kaufen, können Sie APT benutzen, um ein Upgrade auf die neue Version von CD durchzuführen. Dafür benutzen Sie `apt-cdrom` (Siehe Abschnitt 'Hinzufügen einer CD-ROM in die Datei sources.list' auf Seite 9), um die CD zu Ihrer `/etc/apt/sources.list` hinzuzufügen und fahren Sie mit der Installation fort, wie oben beschrieben.

Es ist wichtig zu wissen, dass APT immer nach der aktuellsten Version eines Pakets sucht, aber nur nachdem es die Prioritäten der Quellen berücksichtigt, wie wir in 'APT-Prioritäten für Pakete anpassen' auf Seite 10 gesehen haben. Berücksichtigen Sie das und, falls Ihre `/etc/apt/sources.list` ein Archiv enthält, das eine neuere Version eines Pakets enthält, als die Version auf CD-ROM, würde APT das Paket von dort herunterladen, falls beide Quellen dieselbe Priorität haben, oder das andere Archiv eine höhere Priorität hat, als das auf CD-ROM.

Falls Sie Ihr System nur vom Internet aktualisieren möchen, brauchen Sie nur Ihre Datei `/etc/apt/sources.list` entsprechend aufsetzen. Auf eine neue stabile Veröffentlichung zu aktualisiereni, ist für gewöhnlich nur eine Frage des Durchlaufens des normalen

Aktualisierungs-Prozess in `aptitude`. Falls Ihre `/etc/apt/sources.list` den Kodenamen[4] der Veröffentlichung benutzt, müssen Sie ihn auf den neuen Namen setzen oder in »stable« ändern.

Zum Beispiel würde diese Zeile:

```
deb http://http.us.debian.org/debian sarge main
```

wie eine der folgenden aussehen müssen, sobald »Etch« »stable« wird, damit Ihr Computer auf dem Stand des aktuellen stabilen Systems bleibt:

```
deb http://http.us.debian.org/debian etch main
deb http://http.us.debian.org/debian stable main
```

Der Unterschied hier ist, dass die erste Sie solange bei »etch« hält wie Sie wollen, selbst wenn ein neues »stable« herauskommt. Dies kann nützlich sein, falls Sie selbst entscheiden wollen, wann Sie auf das neue »stable« aktualisieren. Das zweite Beispiel aktualisiert Ihren Rechner automatisch auf die neue Version, wenn sie veröffentlicht wird.

## 5.9 Pakete auf »halten« setzen

Wie wir im vorletzten Abschnitt gesehen haben, markiert `aptitude` automatisch Pakete zur Aktualisierung. Falls Sie aber die aktuelle Version installiert behalten wollen, können Sie es das Paket `auf Halten` setzen lassen.

Dies wird durch Drücken der Taste »=« erreicht. Pakete, die vor Aktualisierungen zurückgehalten werden, sehen so aus:

```
ih    alien                                           8.41      8.41
```

In diesem Beispiel wird das Paket `alien` auf Version 8.41 gehalten, selbst wenn eine neue Version in einer APT-Quelle erscheint – beachten Sie den Buchstaben »`h`« links. Um es wieder aktualisierbar zu machen, markieren Sie es mit »+« zur Installation.

Sie können Pakete auch über die Befehlszeile auf »halten« setzen, indem Sie folgendes ausführen:

```
# aptitude hold paket1 paket2 ...
```

[4] Debian 3.0 nannte sich zum Beispiel »woody«, Debian 3.1 ist »sarge«

## 5.10 Ungenutzte Paket-Dateien entfernen: »clean« und »autoclean«

Wenn ein Paket installiert werden soll, bezieht APT von den Quellen, die in der /etc/apt/sources.list aufgelistet sind, die nötigen Dateien, legt sie in ein lokales Archiv (`/var/cache/apt/archives/`) und fährt mit der Installation fort (siehe 'Installieren und Reinstallieren von Paketen' auf Seite 16).

Nach und nach kann dieses lokale Archiv immer größer werden und eine Menge Platz auf der Festplatte belegen. Auch für diesen Fall bietet APT Werkzeuge an, um sein lokales Archiv zu warten: `APTs` `clean` und `autoclean` Methoden.

Die Methode »clean« wird aufgerufen über den Punkt `Paketcache komplett leeren` aus dem Menü `Aktionen`. Sie wird benutzt um alle heruntergeladenen `.deb`-Dateien zu löschen. Die Methode »autoclean« wird über den Punkt `Nur veraltete Paketdateien löschen`, auch im Menü `Aktionen`, aufgerufen.

Die Methode »autoclean« soll hauptsächlich von denen benutzt werden, die gerne eine lokale Kopie der aktuell installierten Pakete behalten wollen. Sie löscht nur diejenigen `.deb`-Dateien, die nicht mehr von irgendeiner APT-Quelle bereitgestellt werden und somit von einer neueren Version abgelöst wurden.

# Kapitel 6

# Aus den Quellen bauen

## 6.1 Herunterladen von Quellpaketen

In der Welt der freien Software ist es üblich, den Quellcode zu studieren oder auch Korrekturen an fehlerhaftem Code vorzunehmen. Ein Paket aus den Quellen zu bauen kann auch nützlich sein, falls Sie Features aktivieren wollen, die im offiziellen Paket deaktiviert sind, oder welche deaktivieren wollen, die aktiviert sind. Um dieses zu tun, müssen Sie den Quellcode des Programms herunterladen. Das APT-System bietet eine einfache Möglichkeit, den Quellcode der vielen Programme der Distribution einschließlich aller für das Erstellen eines `.deb` des Programms nötigen Dateien zu beziehen.

Eine andere übliche Anwendung für Debian-Quellen ist es eine aktuellere Version eines Programms aus der Distribution »unstable« zum Beispiel in »stable« zu benutzen. Das Paket »gegen« »stable« zu kompilieren erzeugt ein Paket mit Abhängigkeiten, die auf die Pakete aus dieser Distribution ausgerichtet sind.

Hierfür sollte der `deb-src`-Eintrag in Ihrer `/etc/apt/sources.list` auf »unstable« zeigen. Es sollte außerdem aktiviert sein, d.h. eventuelle Kommentarzeichen vor der Zeile müssen entfernt werden (Siehe Abschnitt 'Die Datei /etc/apt/sources.list' auf Seite 7).

`aptitude`, Debians Paketmanager, ist hauptsächlich auf Binär-Pakete ausgerichtet. Um Quellpakete zu benutzen müssen wir stattdessen `apt-get` benutzen. Um ein Quellpaket herunterzuladen, benutzen Sie folgenden Befehl:

```
$ apt-get source Paketname
```

Dies lädt drei Dateien herunter: ein `.orig.tar.gz`, ein `.dsc` und ein `.diff.gz`. Im Falle von Debian-nativen Paketen fällt das letzte weg und das erste hat für gewöhnlich kein »`orig`« im Namen.

Die Datei `.dsc` wird von `dpkg-source` benutzt, um das Quellpaket in das Verzeichnis *Paketname-Version* zu entpacken. In jedem heruntergeladenen Quellpaket befindet sich ein Verzeichnis `debian/`, welches die für das Bauen des `.deb`-Paketes nötigen Dateien enthält.

Um das Paket beim Herunterladen automatisch zu erzeugen, fügen Sie einfach -b zur Befehlszeile hinzu:

```
$ apt-get -b source Paketname
```

Wenn Sie sich dazu entscheiden, das Paket noch nicht beim Herunterladen zu erzeugen, können Sie dieses später nachholen, indem Sie den folgenden Befehl in dem Verzeichnis, in dem das Paket nach dem Herunterladen entpackt wurde, ausführen:

```
$ dpkg-buildpackage -rfakeroot -uc -b
```

Beachten Sie, dass Sie zum Bauen der meisten Pakete zumindest die folgenden Pakete benötigen: devscripts, dpkg-dev, debhelper und fakeroot; werfen Sie einen Blick auf 'Installieren und Reinstallieren von Paketen' auf Seite 16 und installieren Sie diese, bevor Sie fortfahren. Die meisten Pakete hängen auch von Kompilierungs-Werkzeugen ab, so dass es nützlich ist, auch das Paket build-essential zu installieren. Andere Pakete könnten gebraucht werden, werfen Sie einen Blick auf 'Für das Kompilieren eines Quellpaketes notwendige Pakete' auf dieser Seite für weitere Informationen.

Um das von den obigen Befehlen erzeugte Paket zu installieren muss man den Paket-Manager direkt benutzen. Werfen Sie einen Blick auf 'Manuell heruntergeladene oder lokal erzeugte Debian-Pakete installieren' auf Seite 19. Das Paket devscripts enthält ein nützliches Werkzeug: debi. Falls Sie debi innerhalb des Verzeichnisses des Quellpaket ausführen, sucht es nach der .changes-Datei im übergeordneten Verzeichnis, um herauszufinden, welche Binärpakete das Quellpaket erzeugt, und es führt dpkg aus, um diese zu installieren. Dies ist in den meisten Fällen nützlich, außer wenn das Quellpaket kollidierende Binärpakete erzeugt. Natürlich brauchen Sie Root-Rechte, um dies auszuführen.

Es besteht ein Unterschied zwischen apt-gets Methode source und seinen anderen Methoden. Die source-Methode kann von normalen Benutzern ohne Root-Rechte benutzt werden. Die Dateien werden in das Verzeichnis heruntergeladen, aus dem der Befehl apt-get source Paket aufgerufen wurde.

## 6.2 Für das Kompilieren eines Quellpaketes notwendige Pakete

Normalerweise müssen sich spezielle Header-Dateien und Bibliotheken auf dem System befinden, um ein Quellpaket zu kompilieren. Alle Quellpakete haben ein Feld mit Namen »Build-Depends:« in ihrer Kontrolldatei, welches die Namen der zusätzlichen Pakete enthält, die für das Erzeugen des Paketes aus dem Quellcode nötig sind. Einige grundlegende Pakete werden ebenfalls benötigt, siehe 'Herunterladen von Quellpaketen' auf der vorherigen Seite, bevor Sie fortfahren.

APT bietet eine einfache Möglichkeit diese Pakete herunterzuladen. Führen Sie einfach apt-get build-dep Paket aus, wobei »Paket« für den Namen des Pakets, welches Sie erzeugen wollen, steht. Beispiel:

```
# apt-get build-dep gmc
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Die folgenden NEUEN Pakete werden installiert:
  comerr-dev e2fslibs-dev gdk-imlib-dev imlib-progs libgnome-dev libgnorba-de
  libgpmg1-dev
0 aktualisiert, 7 neu installiert, 0 zu entfernen und 1 nicht aktualisiert.
Es müssen 1069kB Archive geholt werden. Nach dem Auspacken werden 3514kB
benutzt.
Möchten Sie fortfahren? [J/n]
```

Die Pakete, die installiert werden, werden gebraucht, um `gmc` korrekt zu erzeugen. Es ist wichtig zu beachten, dass dieser Befehl nicht das Quellpaket des zu bauenden Programms herunterlädt. Sie müssen hierfür zusätzlich `apt-get source` ausführen.

Falls Sie nur feststellen möchten, welche Pakete zum Bau eines bestimmten Paketes benötigt werden, gibt es eine Methode `showpkg` des Befehls `apt-cache` (siehe 'Informationen über Pakete bekommen' auf Seite 33), die neben anderer Informationene die Zeile `Build-Depends` aufführt, die ihrerseits die erforderlichen Pakete auflistet.

```
# apt-cache showsrc Paket
```

## 6.3 Ein Paket zur Fehlersuche bauen

Wenn Sie ein Paket zum Zwecke der Fehlersuche bauen wollen, zum Beispiel um einen Fehler zu melden – oder zu reparieren, können Sie einfache Umgebungsvariablen benutzen, die von den meisten Debian-Paketen unterstützt werden.

Um ein Paket zu bauen, das ungestrippte[1] Binärdateien enthält, müssen Sie dem Bau-Befehl nur `DEB_BUILD_OPTIONS=nostrip` voranstellen. Optimierungen können ebenfalls die Fehlersuche erschweren, deshalb können Sie sie durch Hinzufügen von `noopt` zu der Variablen DEB_BUILD_OPTIONS ebenfalls deaktivieren. Ein Beispiel:

```
$ DEB_BUILD_OPTIONS="nostrip noopt" dpkg-buildpackage -rfakeroot -uc -b
```

## 6.4 Anpassen von Optionen des Paket-Baus

Wenn Sie die Art und Weise anpassen wollen, wie das Paket gebaut wird, müssen Sie daran gehen, die Datei `debian/rules` zu editieren. Dies heißt: Innerhalb des Hauptverzeichnisses, das erzeugt wird wenn das Quellpaket entpackt wird, ist ein Verzeichnis `debian`, das viele Dateien enthält. Eine davon ist besonders: Die Datei `rules`.

[1] Debian strippt für gewöhnlich alle ausgelieferten Binärdateien und Bibliotheken. Das heißt, dass alle Symbole zur Fehlersuche entfernt werden.

Diese Datei ist für gewöhnlich ein Makefile, das Ziele hat zum Konfigurieren, Bauen, Installieren und Erzeugen des Pakets. Falls Sie zum Beispiel das Paket `luola`[2] ohne Sound bauen wollen, kann ich, nachdem ich seine Quellen heruntergeladen und entpackt habe, die Datei `debian/rules` editieren, die wie folgt aussieht:

```
[...]
configure: configure-stamp
configure-stamp:
 dh_testdir
 # Add here commands to configure the package.
 ./configure $(confflags) \
 --prefix=/usr \
 --mandir=share/man \
 --infodir=share/info \
 --datadir=share/games \
 --bindir=games \
 --enable-sound
#  --enable-sdl-gfx

 touch configure-stamp
[...]
```

Sehen Sie den Schalter `--enable-sound`? Falls ich ihn entferne oder durch `--disable-sound` ersetze und dann das Paket unter Benutzung der in 'Herunterladen von Quellpaketen' auf Seite 25 beschriebenen Technik neu baue, habe ich ein luola-Paket, das ohne Sound gebaut wurde.

Wenn Sie wirklich auf einer täglichen Basis mit Quellpaketen arbeiten möchten, würde ich hauptsächlich das Lesen des Debian New Maintainers Guide (http://www.debian.org/doc/maint-guide/) (Führer für neue Debian-Betreuer) und des Debian Policy (http://www.debian.org/doc/debian-policy/) (Debian-Richtlinien-Handbuch) vorschlagen. Andere in der Debian Developers Corner (http://www.debian.org/devel/) (Debians Entwicklerecke) verfügbare Dokumentation mag ebenfalls nützlich sein.

## 6.5 Aber heh, ich will den Debian-Kram nicht benutzen!

Manchmal will man spezielle Versionen eines Programms benutzen, die nur als Quellcode verfügbar sind und nicht als Debian-Paket. Hier kann es allerdings Probleme mit dem Paket-System geben. Angenommen Sie wollen eine neue Version Ihres Mailservers kompilieren und alles klappt, aber viele Pakete in Debian hängen von einem MTA (Mail Transfer Agent) ab. Da etwas installiert wurde, was Sie selbst kompiliert haben, weiß das Paketsystem darüber nicht Bescheid.

[2] Ein sehr nettes Kampfspiel mit Schiffen, dem ein Netzwerk-Modus fehlt. Irgendwelche Mitspieler?

Hier kommt das `equivs` ins Spiel. Um es zu benutzen, installieren Sie das Paket mit diesem Namen. Es erstellt ein leeres Paket, das die Abhängigkeiten erfüllt und dem Paketsystem mitteilt, so dass es keine Probleme mit Abhängigkeiten gibt.

Bevor wir näher darauf eingehen, ist es wichtig, Sie darauf hinzuweisen, dass es sicherere Möglichkeiten gibt, Programme, für die in Debian schon Pakete existieren, mit anderen Optionen zu kompilieren und man equivs nicht benutzen sollte, um Abhängigkeiten zu entfernen, ohne genau zu wissen, was man tut. Siehe 'Herunterladen von Quellpaketen' auf Seite 25 für nähere Informationen.

Lassen Sie uns mit dem MTA-Beispiel fortfahren. Sie haben also gerade Ihren frisch kompilierten `postfix` installiert und wollen nun `mutt` (ein Mailprogramm) installieren. Plötzlich stellen Sie fest, dass `mutt` einen anderen MTA installieren möchte, obwohl Sie schon Ihren selbstkompilierten laufen haben.

Wechseln Sie in irgendein Verzeichnis (z.B. `/tmp`), und führen Sie folgendes aus:

```
# equivs-control Name
```

Ersetzen Sie *Name* durch den Namen der Kontrolldatei, die Sie erstellen wollen. Die Datei wird wie folgt erstellt:

```
Section: misc
Priority: optional
Standards-Version: 3.0.1

Package: <Paketname; wenn nicht angegeben: equivs-dummy>
Version: <Versionsnummer; wenn nicht angegeben: 1.0>
Maintainer: <Ihr Name mit E-Mail-Adresse; wenn nicht angegeben: Benutzername
Pre-Depends: <Pakete>
Depends: <Pakete>
Recommends: <Pakete>
Suggests: <Pakete>
Provides: <(virtuelles) Paket>
Architecture: all
Copyright: <copyright Datei; normalerweise GPL2>
Changelog: <changelog file; normalerweise ein generisches Changelog>
Readme: <README.Debian file; wenn nicht angegeben, ebenfalls ein generisches>
Extra-Files: <Zusätzliche Dateien für das doc-Verzeichnis, durch Komma getren
Description: <kurze Beschreibung; Standard ist "some wise words">
 Lange Beschreibung und Info
 .
 Zweiter Absatz
```

Nun muss das so angepaßt werden, dass es tut, was wir wollen. Gucken Sie sich die Felder und ihre Beschreibungen an, es ist nicht nötig, jedes einzelne hier zu erklären, lassen Sie uns stattdessen ein Beispiel ansehen:

```
Section: misc
Priority: optional
Standards-Version: 3.0.1

Package: mta-local
Conflicts: mail-transport-agent
Replaces: mail-transport-agent
Provides: mail-transport-agent
```

Das war es schon. `mutt` hängt von `mail-transport-agent` ab, was ein virtuelles Paket ist, was alle MTAs liefern; mta-local meldet sich selbst unter Benutzung des Feldes »Provides« (Liefert) als ein mail-transport-agent an.

Die Felder `Conflicts` (Kollidiert) und `Replaces` (Ersetzt) werden ebenfalls benötigt, so dass `APT` und `dpkg` wissen, dass sie das aktuell installiert MTA-Paket zugunsten des neuen, das Sie installieren, entfernen sollten.

Nun muss das Paket nur noch gebaut werden:

```
# equivs-build name
dh_testdir
touch build-stamp
dh_testdir
dh_testroot
dh_clean -k
# Add here commands to install the package into debian/tmp.
touch install-stamp
dh_testdir
dh_testroot
dh_installdocs
dh_installchangelogs
dh_compress
dh_fixperms
dh_installdeb
dh_gencontrol
dh_md5sums
dh_builddeb
dpkg-deb: Baue Paket »Name« in »../Name_1.0_all.deb«.

The package has been created.
Attention, the package has been created in the current directory,
```

Und installieren Sie das erzeugte `.deb` (siehe ‘Manuell heruntergeladene oder lokal erzeugte Debian-Pakete installieren’ auf Seite 19.

Wie man erkennen kann, gibt es verschiedene Anwendungen für `equivs`. Man könnte sogar ein `Favoriten`-Paket erstellen, was von den Paketen abhängt, die Sie normalerweise installieren. Lassen Sie Ihren Vorstellungen einfach freien Lauf, aber seien Sie vorsichtig.

Es ist wichtig zu erwähnen, dass es in `/usr/share/doc/equivs/examples` einige Beispiel-Kontrolldateien gibt. Werfen Sie mal einen Blick darauf.

# Kapitel 7

# Informationen über Pakete bekommen

Es gibt einige Front-End Programme für das APT-System, die es signifikant einfacher machen, Listen über verfügbare oder schon installierte Pakete zu bekommen oder auch herauszufinden, zu welcher Sektion ein Paket gehört, welche Priorität es hat, wie seine Beschreibung lautet, usw.

Unser Ziel aber hier ist, APT selbst benutzen zu lernen. Wie können wir also den Namen eines Paketes herausfinden, welches wir installieren wollen?

Dies ist es, was diese Kapitel beantworten möchte. Lassen Sie uns unsere Optionen austesten.

## 7.1 Paketnamen entdecken

Angenommen, Sie wollen die alten Zeiten des Atari 2600 wieder aufleben lassen. Sie möchten APT benutzen, um einen Atari-Emulator zu installieren und dann Spiele herunterladen. Sie haben folgende Möglichkeit:

```
$ aptitude search atari
p   atari-fdisk-cross                - Partition editor for Atari (running on
p   atari800                         - Atari emulator for X/curses/SDL
p   console-keymaps-atari            - Keymaps for Atari keyboards.
```

Wir finden einige Pakete, die zu dem, was wir suchen, in Beziehung stehen, zusammen mit kurzen Beschreibungen. Der Buchstabe »p« am Zeilenanfang ist ähnlich der Notation, die in `aptitude`s Curses Benutzerschnittstelle benutzt wird. Er bedeutet also, dass das Paket nicht installiert ist. Es ist wichtig zu beachten, dass `aptitude` standardmäßig nur Paketnamen durchsucht. Sie können alle Such-Optionen benutzen, die `aptitude` bereitstellt, die Sie im Benutzerhandbuch finden können. Um zum Beispiel die Beschreibungen zu durchsuchen, können Sie folgendes benutzen:

```
$ aptitude search ~datari
```

```
p   aranym                          - Atari Running on Any Machine
p   atari-fdisk-cross               - Partition editor for Atari (running on
p   atari800                        - Atari emulator for X/curses/SDL
p   circuslinux                     - The clowns are trying to pop balloons t
p   circuslinux-data                - Data files for circuslinux
p   console-keymaps-atari           - Keymaps for Atari keyboards.
[...]
p   stella                          - Atari 2600 Emulator for SDL & X windows
[...]
```

Wie wir sehen, werden jetzt viel mehr Pakete zurückgegeben. Nun, da wir eine Liste möglicher Lösungen für unser Problem gefunden haben, lassen Sie uns einen Schritt tiefer gehen:

```
$ aptitude show stella
Paket: stella
Zustand: nicht installiert
Version: 2.2-1
Priorität: optional
Bereich: contrib/otherosfs
Verwalter: Mario Iseli <admin@marioiseli.com>
Unkomprimierte Größe: 2998k
Hängt ab von: libc6 (>= 2.3.5-1), libgcc1 (>= 1:4.1.0), libpng12-0 (>=
              1.2.8rel), libsdl1.2debian (> 1.2.7+1.2.8), libstdc++6 (>= 4.1.
              zlib1g (>= 1:1.2.1)
Beschreibung: Atari 2600 Emulator for SDL & X windows
 Stella is a portable emulator of the old Atari 2600 video-game console writt
 in C++.  You can play most Atari 2600 games with it.

 Homepage: http://stella.sf.net/
```

Mit dieser Ausgabe erhalten Sie eine Menge Details über das Paket, das Sie installieren wollen (oder nicht wollen) inklusive der vollständigen Beschreibung des Pakets. Sie können auch das Programm apt-cache benutzen, um solche Informationen zu bekommen. Falls Sie dieses Werkzeug benutzen, ist es fähig mehrere verfügbare Versionen eines Paketes anzuzeigen, zum Beispiel:

```
$ apt-cache show muine
Package: muine
Version: 0.5.0-1
Priority: optional
Section: gnome
Maintainer: Link Dupont <link@subpop.net>
Depends: gstreamer-gconf (>= 0.6.4), libatk1.0-0 (>= 1.4.1), libbonobo2-0 (>=
Architecture: i386
Filename: ./pool/main/m/muine/muine_0.5.0-1_i386.deb
```

```
Size: 164314
Installed-Size: 692
MD5sum: 9885f13e5ef4f76b3bf6fe7bb3ea8634
Description: Simple music player
 Muine is an innovative music player. It has a simple interface designed to
 allow the user to easily construct playlists from albums and/or single songs
 Its goal is to be simply a music player, not to become a robust music
 management application.

Package: muine
Priority: optional
Section: gnome
Installed-Size: 492
Maintainer: Link Dupont <link@subpop.net>
Architecture: i386
Version: 0.4.0-8
Depends: gstreamer-gconf (>= 0.6.4), libatk1.0-0 (>= 1.4.1), libbonobo2-0 (>=
Filename: pool/main/m/muine/muine_0.4.0-8_i386.deb
Size: 139864
MD5sum: e746709ad6a6fcc1e9c46b46b3d6e5b2
Description: Simple music player
 Muine is an innovative music player. It has a simple interface designed to
 allow the user to easily construct playlists from albums and/or single songs
 Its goal is to be simply a music player, not to become a robust music
 management application.
```

Für allgemeinere Informationen über ein Paket können Sie folgendes benutzen:

```
# apt-cache showpkg penguin-command
Package: penguin-command
Versions:
1.4.5-1(/var/lib/apt/lists/download.sourceforge.net_debian_dists_unstable_mai

Reverse Depends:
Dependencies:
1.4.5-1 - libc6 (2 2.2.1-2) libpng2 (0 (null)) libsdl-mixer1.1 (2 1.1.0) libs
Provides:
1.4.5-1 -
Reverse Provides:
```

Und um nur herauszufinden, von welchen Paketen es abhängt:

```
# apt-cache depends penguin-command
penguin-command
  Depends: libc6
```

```
Depends: libpng2
Depends: libsdl-mixer1.1
Depends: libsdl1.1
Depends: zlib1g
```

Zusammengefaßt haben wir eine handvoll Waffen, die wir benutzen können, um den Namen des Paketes herauszufinden, das wir installieren wollen.

## 7.2 Paketnamen mit dpkg finden

Ein Weg, den Namen eines Pakets zu finden, ist, den Namen einer wichtigen Datei zu kennen, die sich in dem Paket befindet. Um zum Beispiel das Paket zu finden, welches eine bestimmte ».h« Datei enthält, die für das Kompilieren eines Programms benötigt wird, ist folgendes auszuführen:

```
# dpkg -S stdio.h
libc6-dev: /usr/include/stdio.h
libc6-dev: /usr/include/bits/stdio.h
perl: /usr/lib/perl/5.6.0/CORE/nostdio.h
```

oder:

```
# dpkg -S /usr/include/stdio.h
libc6-dev: /usr/include/stdio.h
```

Beachten Sie, dass diese Methode nur Paketnamen von Paketen findet, die auf Ihrem System installiert sind. Sie müssen auto-apt (siehe 'Wie Pakete »nach Bedarf« installieren' auf Seite 39) oder apt-file (siehe 'Herausfinden, zu welchem Paket eine Datei gehört' auf der nächsten Seite) benutzen, um nach Dateien in nicht-installierten Paketen zu suchen. Sie können auch einen Blick auf das Webangebot http://packages.debian.org/ werfen. Um die Namen der auf Ihrem System installierten Pakete aufzulisten, was zum Beispiel nützlich ist, falls Sie planen, Ihre Festplatte aufzuräumen, können Sie folgendes ausführen:

```
# dpkg -l | grep mozilla
ii  mozilla-browse 0.9.6-7        Mozilla Web Browser
```

Das Problem mit diesem Befehl ist, dass er Paketnamen »brechen« kann. Im obigen Beispiel ist der ganze Name des Pakets mozilla-browser. Um das Problem zu beheben, können Sie die Umgebungsvariable COLUMNS folgendermaßen benutzen:

```
[kov]@[couve] $ COLUMNS=132 dpkg -l | grep mozilla
ii  mozilla-browser             0.9.6-7                      Mozilla Web Brows
```

## 7.3 Herausfinden, zu welchem Paket eine Datei gehört

Wenn ein Paket installiert werden soll und Sie nicht herausfinden können, wie es heißt, indem Sie mit `apt-cache` suchen, aber den Dateinamen des Programms oder einer Datei, die zu dem Paket gehört kennen, können Sie `apt-file` benutzen, um den Dateinamen zu finden. Das wird folgendermaßen gemacht:

```
$ apt-file search Dateinamen
```

Es funktioniert genau wie `dpkg -S`, es zeigt Ihnen aber auch nicht installierte Pakete, die die Datei enthalten. Man kann es auch dazu benutzen, benötigte include-Dateien, die beim Kompilieren von Programmen fehlen, zu installieren, allerdings ist `auto-apt` eine wesentlich bessere Methode solche Fälle zu lösen, siehe 'Wie Pakete »nach Bedarf« installieren' auf Seite 39.

Man kann auch den Inhalt von Paketen auflisten:

```
$ apt-file list Paketname
```

`apt-file` hat genau wie auto-apt eine Datenbank über die Dateien aller Pakete und diese muss aktuell gehalten werden. Dies wird erledigt durch:

```
# apt-file update
```

Standardmäßig benutzt `apt-file` die gleiche Datenbank wie `auto-apt`, siehe 'Wie Pakete »nach Bedarf« installieren' auf Seite 39.

## 7.4 Informationen über Pakete im Web

Es gibt viele Web-Resourcen mit Informationen über die in der Debian-Distribution verfügbaren Pakete, die meisten richten sich hauptsächlich an Debian-Entwickler, aber viele von ihnen können auch für Benutzer nützlich sein.

Die Debian Packages Pages (`http://packages.debian.org/`) (Paket-Seiten) lassen Sie nach Paketen suchen, die für die verschiedenen Architekturen, die Debian unterstützt, verfügbar sind, und auch nach dem Inhalt aller Debian-Pakete. Es gibt Informationen bezüglich Abhängigkeiten und anderer Beziehungengen mit anderen Paketen. Es gibt ebenso Links um Quell- und Binärpaket für alle Architekturen herunter zu laden. Sie können einen Shortcut `http://packages.debian.org/`*`Paketname`* benutzen, um schnellen Zugriff auf ein Paket zu bekommen.

Das Package Tracking System (`http://packages.qa.debian.org/`) (Paket-Verfolgungs-System) liefert Informationen darüber, was kürzlich mit einem Paket geschehen ist, was

die zu erledigenden Punkte aus der Sicht eines Debian-Betreuers sind, liefert eine Zusammenfassung der Fehlerberichte, und einige mehr sehr nützliche Informationen. Eines der netten Dinge des Paket-Verfolgungs-Systems ist die Möglichkeit, sich für ein Paket »einzuschreiben«, um alle E-Mails zu verfolgen, die gewöhnlich der Betreuer des Paketes bekommt. Auf diese Weise können Sie die Entwicklung der Pakete verfolgen, die entscheidend sind für Ihre Arbeit oder Ihr Hobby. Auch hier funktioniert der Shortcut `http://packages.qa.debian.org/`*`Paketname`*.

Schließlich liefert das Bug Tracking System (http://bugs.debian.org/) (Fehler-Verfolgungs-System) Informationen über bekannte Fehler in den Paketen, die von Debian verteilt werden. Dies mag für Sie nützlich sein, um herauszufinden, warum etwas nicht so funktioniert, wie Sie erwarten, und sogar um Lösungen und Work-Arounds in den Fehler-Aufzeichnungen zu finden. Lesen Sie diese Seiten und ihre Dokumentation auch, um in der Lage zu sein, gute Fehlerberichte für Debian einreichen zu können. Der Shortcut `http://bugs.debian.org/`*`Paketname`* funktioniert auch hier wie erwartet, aber das BTS (Bug Tracking System) akzeptiert auch andere sehr nützliche Shortcuts, wie zum Beispiel `http://bugs.debian.org/src:`*`Paketname`* um alle Fehlerberichte aller Binärpakete, die vom angebenen Quellpaket geliefert werden, anzuzeigen.

# Kapitel 8

# Tricks und Techniken

## 8.1 Wie Pakete »nach Bedarf« installieren

Sie kompilieren gerade ein Programm, und wie aus heiterem Himmel - Boom! Es gibt einen Fehler, da eine `.h` Datei gebraucht wird, die Sie nicht haben. Das Programm `auto-apt`, enthalten im Paket desselben Namens, kann Sie vor solchen Szenarios bewahren. Es fragt, ob es die benötigten Pakete installieren soll, nachdem es den betroffenden Prozess gestoppt hat und führt ihn fort, wenn die relevanten Pakete installiert sind.

Der Befehl sieht folgendermaßen aus:

```
# auto-apt run Befehl
```

Wobei *Befehl* der Befehl ist, der ausgeführt werden soll und evtl. nicht vorhandene Dateien benötigt. Beispiel:

```
# auto-apt run ./configure
```

Es wird fragen, ob die benötigten Pakete installiert werden sollen, und `apt-get` automatisch aufrufen. Wenn X läuft, ersetzt eine grafische Oberfläche die übliche Text-Oberfläche.

Auto-apt funktioniert mit einer Datenbank, welche aktuell gehalten werden muss, um effektiv zu funktionieren. Das erreicht man mit den Kommandos `auto-apt update`, `auto-apt updatedb` und `auto-apt update-local`.

## 8.2 Pakete zwischenspeichern

Falls Sie mehrere Maschinen in einem Netzwerk haben, wäre es nützlich ein System zur Paketzwischenlagerung zu haben, so dass Sie die Pakete nicht für jede Maschine herunterladen müssen. Diese Anleitung befasst sich mit dem Paket `apt-proxy`, das genau dies tut. Es gibt

jedoch andere Lösungen, wie apt-cacher. Sie mögen jede ausprobieren wollen, um die auszuwählen, die am besten Ihren Bedürfnissen entspricht.

Zuallererst installieren Sie das Paket apt-proxy. Es registriert sich beim inetd, um auf Anfragen auf Port 9999 zu warten, Sie mögen den Dienst inetd neustarten müssen.

Als nächstes werden Sie die Datei /etc/apt-proxy/apt-proxy.conf editieren wollen. Sie listet die wahren Server auf, von denen apt-proxy Paketlisten und Pakete herunterlädt. Sie können http, ftp und rsync als Transfermethoden benutzen. Die Datei hat ein Standard-»Backend«, das wie folgt aussieht:

```
add_backend /debian/                                    \
        $APT_PROXY_CACHE/debian/                        \
        http://ftp.us.debian.org/debian/                \
        http://ftp.de.debian.org/debian/                \
        http://ftp.uk.debian.org/debian/                \
        +ftp.us.debian.org::debian/
```

Dies heißt, dass wann immer ein Client versucht, etwas aus /debian/ zu bekommen, benutzt apt-proxy seinen Zwischenspeicher, die aufgelisteten HTTP-Spiegel und einen RSYNC-Server, von dem bevorzugt die Paketlisten heruntergeladen werden (Das Zeichen »+« bedeutet bevorzugt für Kontroll-Dateien).

Wie benutzt nun ein Client dieses Backend? Durch Hinzufügen folgender Zeile zur Datei /etc /apt/sources.list auf den Clients (einschließlich des Rechners, auf dem apt-proxy installiert ist):

```
deb http://server:9999/debian/ stable main
```

Die Zeile sieht sehr wie eine normal Zeile aus. Der Unterschied hier ist, dass Sie dort Ihren apt-proxy-Server setzen, wo Sie normalerweise einen HTTP- oder FTP-Spiegel setzen, dass Sie einen Port-Wert benutzen (9999) und dann das Backend auswählen (/debian/). Nachdem Sie all diese anfänglichen Einstellungen erledigt haben, updaten Sie die Paketliste auf einer der Maschinen und aktualisieren Sie sie, so dass sie nur ein Paket gleichzeitig herunterlädt. Nachdem die erste Maschine auf den aktuellen Stand gebracht wurde, aktualisieren Sie die anderen.

Sie können den Befehl apt-proxy-import benutzen, um die Pakete zu importieren, die sich in Ihrem aktuellen APT-Zwischenspeicher (/var/cache/apt/archives/ befinden, indem Sie apt-proxy-import /var/cache/apt/archives ausführen. Beachten Sie, dass Sie auf mindestens einem Client den Aktualisierungsprozeß durchführen müssen, um apt-proxys Zwischenspeicher zu initialisieren, bevor Sie apt-proxy-import benutzen.

Sie können mehr über apt-proxy erfahren, indem Sie die Kommentare in der Datei /etc /apt-proxy/apt-proxy.conf lesen. Um zum Beispiel apt-proxy mit HTTP- und FTP-Proxies zu benutzen, finden Sie am Ende der Datei Beispielkonfigurationen.

## 8.3 Entscheidung - Welcher Spiegel ist der beste für die sources.list: netselect, netselect-apt

Eine häufige Frage der meist neuen Benutzer ist: »Welchen Debian-Spiegel soll ich in die `sources.list` eintragen?«. Es gibt viele Wege, sich für einen Mirror zu entscheiden. Die Experten haben möglicherweise ein Skript, welches den Ping der verschiedenen Spiegel vergleicht. Aber es gibt ein Programm, das dies für uns erledigt: `netselect`.

Man installiert `netselect` wie üblich:

```
# aptitude install netselect
```

Wenn man es ohne Parameter ausführt, zeigt es seinen Hilfetext an. Führt man es mit einer durch Leerzeichen separierten Liste von Hostnamen (Spiegeln) aus, gibt es uns einen Hostnamen zusammen mit der einer Punktzahl zurück. Diese Punktzahl berücksichtigt die erwarteten Pingzeit und die Zahl der Hops (Rechner, die eine Netzwerkanfrage passiert, um ihren Zielort zu erreichen) und ist antiproportional zur erwarteten Downloadgeschwindigkeit (also je niedriger, desto besser). Angezeigt wird nur der Host mit der niedrigsten Punktzahl (Die ganze Liste der Mirrors kann mit der Option -vv angesehen werden). Zum Beispiel:

```
# netselect ftp.debian.org http.us.debian.org ftp.at.debian.org download.unes
  365 ftp.debian.org.br
#
```

Das bedeutet, dass von den Mirrors die als Parameter an netselect übergeben wurden, `ftp.debian.org.br` der beste war mit einer Punktzahl von 365. (Achtung! Weil es auf meinem Computer ausgeführt wurde und die Netzwerktopographie extrem unterschiedlich und abhängig vom Standort des Computers ist, ist dieser Wert nicht notwendigerweise die richtige Geschwindigkeit für andere Computer).

Jetzt tragen Sie einfach den schnellsten Spiegel in die Datei `/etc/apt/sources.list` ein (siehe 'Die Datei /etc/apt/sources.list' auf Seite 7) und befolgen Sie die Tipps in 'Paketverwaltung' auf Seite 13.

**Hinweis:** Die Liste der Spiegel ist immer auf `http://www.debian.org/mirror/mirrors_full` zu finden.

Ab Version 0.3 enthält das netselect-Paket das **netselect-apt**-Script, das obigen Prozess automatisiert. Übergeben Sie einfach die Distribution als Parameter (Der Standardwert ist stable) und die `sources.list` wird mit den besten »main«- Spiegeln generiert und im aktuellen Verzeichnis gespeichert. Das folgende Beispiel generiert eine sources.list für die stabile Distribution:

```
# ls sources.list
ls: sources.list: File or directory not found
# netselect-apt stable
```

```
(...)
# ls -l sources.list
sources.list
#
```

**Hinweis:** Die `sources.list` wird im aktuellen Verzeichnis erzeugt und muss nach `/etc/apt` verschoben werden.

## 8.4 Entfernen von unbenutzten locale-Dateien: localepurge

Viele Debian-Benutzer verwenden nur ein locale (Spracheinstellung). Ein brasilianischer Debian-Benutzer benutzt z.B. vermutlich immer das brasilianische `pt_BR`-locale und interessiert sich nicht für das spanische `es`-locale.

`localepurge` ist ein sehr nützliches Werkzeug für diese Art von Benutzern. Sie können eine Menge Festplattenplatz sparen, wenn Sie nur die locales installiert haben, die Sie auch wirklich brauchen. Installieren Sie einfach `apt-get install localepurge`.

Es ist wirklich einfach zu konfigurieren, Debconf-Fragen führen den Benutzer Schritt für Schritt durch die Konfiguration. Seien Sie vorsichtig beim Beantworten der ersten Frage, da falsche Antworten alle locales entfernen können - selbst die, die Sie benutzen. Die einzige Möglichkeit, sie wiederherzustellen, ist, alle Pakete neu zu installieren, die sie enthalten.

## 8.5 Wie über Änderungen in Paketen informiert bleiben

Jedes Paket installiert in sein Dokumentationsverzeichnis (`/usr/share/doc/Paketname`) eine Datei mit Namen `changelog.Debian.gz`, welche die Liste der Änderungen gegenüber der letzten Version enthält. Sie können diese Dateien z.B. mit Hilfe von `zless` lesen, aber es ist nicht wirklich leicht, nach einem System-Upgrade nach dem changelog jedes aktualisierten Paketes zu suchen.

Es gibt aber eine Möglichkeit, diese Aufgabe zu automatisieren mit Hilfe eines Werkzeugs mit Namen `apt-listchanges`. Hierfür muss das Paket `apt-listchanges` erst einmal installiert werden. Während der Installation übernimmt Debconf die Installation. Beantworten Sie die Fragen nach Ihren Bedürfnissen.

Die Option »Soll apt-listchanges automatisch von APT gestartet werden?« ist sehr nützlich, weil es eine Liste von Veränderungen an jedem Paket, das von APT während einer Aktualisierung installiert wird, und läßt Sie sie analysieren, bevor Sie fortfahren.Die Option "Soll apt-listchanges nach dem Anzeigen der Changelogs um eine Bestätigung bitten?" ist sehr nützlich, da es eine Liste der Änderungen jedes Paketes, das während eines Upgrades installiert wird, anzeigt und Ihnen die Möglichkeit bietet, diese vor dem Fortfahren einzusehen. Wenn Sie hier sagen, dass Sie nicht fortfahren möchten, gibt apt-listchanges einen Fehlercode zurück und APT bricht die Installation ab.

Nachdem apt-listchanges installiert wurde, zeigt es die Liste der Änderungen installierter Pakete an, wenn Pakete aus dem Netz (oder von einer CD oder gemounteten Partition) heruntergeladen werden, bevor Sie installiert werden.

# Kapitel 9

# Auf der anderen Seite: ein Repository erzeugen

Um ein Repository zu erzeugen, brauchen Sie hauptsächlich eine Reihe von Paketen (Quellen oder Binär) um aus ihnen die Dateien `Packages.gz` und `Sources.gz` zu erzeugen. In einigen Fällen mögen Sie nur Binärpakete haben, aber seien Sie vorsichtig, keine Lizenzen zu verletzen. GPL-lizensierte Software zum Beispiel muss neben den Binärdateien zusammen mit seinen Quellen verteilt werden.

Um das zu erledigen, benutzen Sie das Programm `apt-ftparchive`, das vom Paket `apt-utils` installiert wird. Zuerst sammeln Sie all Ihre Pakete in einem Verzeichnis. Sagen wir `~/public_html/debian`, zum Beispiel; wenn wir dies benutzen, stellen wir bereits unser Repository durch unseren HTTP-Server bereit!

Ich kann in dieses Verzeichnis gehen und folgendes ausführen:

```
$ apt-ftparchive sources . > Sources
$ apt-ftparchive packages . > Packages
$ gzip -c Packages > Packages.gz
$ gzip -c Sources > Sources.gz
```

Die ersten beiden Befehle durchsuchen das angebene Verzeichnis (in diesem Fall `.`) und seine Unterverzeichnisse und geben die entstehenden Paketlisten auf der Standardausgabe aus, die wir in die Dateien `Sources` und `Packages` umleiten.

Falls Sie die `file`-Methode benutzen, um auf das Repository zuzugreifen, sind die ersten beiden Zeilen genug. Falls Sie aber eine Methode für entfernten Zugriff benutzen, wie `FTP` oder `HTTP`, brauchen Sie die komprimierten Versionen, weil APT danach sucht, wenn es die Paketlisten updatet.

Hier sind die Quellzeilen, die Sie Ihrer `/etc/apt/sources.list` hinzufügen können, um APT Ihr Repository bekannt zu machen:

```
# Dies funktioniert nur bei lokalem Zugriff auf das Dateisystem
```

```
deb file:/home/Benutzer/public_html/debian/ ./

# Dies kann von Leuten benutzt werden, die auf Ihre
# Maschine über eine Netzwerkverbindung zugreifen können,
# falls Sie Ihren HTTP-Server entsprechen konfiguriert haben
deb http://Host/~Benutzer/debian/ ./
```

Sie bekommen eine Vorstellung. So wie wir `apt-ftparchive` imselben Verzeichnis, in dem die Pakete lagen, können wir einfach den vollen Pfad dahin, wo die Pakete sind, angeben und ein `./` am Ende anfügen. Sie können es anders machen, indem Sie `apt-ftparchive` mit anderen relativen Pfaden aufrufen, statt es in demselben Verzeichnis, in dem die Pakete liegen, aufzurufen. Sie können auch Gruppen von Paketen in Sektionen trennen, wie es Debian tut. Lesen Sie die Manpage `apt-ftparchive(1)` und spielen Sie mit den Möglichkeiten.

# Kapitel 10

# Alternative Benutzerschnittstellen zur Paketverwaltung

Dieses Howto empfiehlt aptitude als die Haupt-Benutzerschnittstelle zur Paketverwaltung Ihres Debian-Systems. Dies ist, weil, wie wir in diesem Dokument gesagt haben, aptitude eine Reihe von Featuren hat, die das Leben des Administrators einfacher machen.

Es gibt jedoch andere Benutzerschnittstellen, sogar grafische, wie Synaptic oder GNOME-APT. In diesem Kapitel werden wir sie durchnehmen.

## 10.1 Synaptic

Synaptic wurde ursprünglich von Alfredo Kojima für Conectiva geschrieben. Heute wird es von Leute aus Conectiva und Debian betreut. Es wurde ursprünglich für Wing, Window Makers grafisches Toolkit, geschrieben, wurde aber für GTK+ umgeschrieben.

Um Synaptic zu benutzen, müssen Sie das Paket `synaptic` installieren.

Synaptic ist wahrscheinlich die am besten benutzbare aller grafischen Benutzerschnittstellen für APT, die mit der Endbenutzer die wenigsten Probleme haben.

## 10.2 GNOME-APT

GNOME-APT wurde zuerst von Havoc Pennigton und Diego Lages für Debian geschrieben. Es blieb einige Zeit unbetreut, aber die Leute haben es geschafft, es für Woody am Leben zu halten. Für Sarge hat das Paket einen neuen Betreuer bekommen, der es weiterbrachte. Im Gegensatz zu Synaptic ist GNOME-APT für die Leute, die, während sie sich an einer schönen GTK+ Schnittstelle erfreuen, Flexibilität Benutzbarkeit vorziehen.

Sie können GNOME-APT benutzen, indem Sie das Paket `gnome-apt` installieren.

# Kapitel 11

# Der Umgang mit Fehlern

## 11.1 Häufige Fehler

Fehler wird es immer geben. Viele werden durch unachtsame Benutzer verursacht. Im Folgenden finden Sie eine Liste mit häufig gemeldeten Fehlern und wie Sie mit ihnen umgehen sollten.

Wenn Sie eine Nachricht erhalten, die aussieht wie die im unteren Beispiel, bei dem Versuch `apt-get install Paket` auszuführen ...

```
Paketlisten werden gelesen... FertigAbhängigkeitsbaum wird aufgebaut... Ferti
W: Probieren Sie »apt-get update«, um diese Probleme zu korrigieren.
E: Konnte Paket penguineyes nicht finden
```

haben Sie vergessen `apt-get update` nach Ihrer letzten Änderung in der `/etc/apt/sources.list` auszuführen.

Wenn folgender Fehler auftritt ...

```
E: Konnte Lockdatei /var/lib/dpkg/lock nicht öffnen - open (13 Keine Berechti
E: Unable to lock the administration directory (/var/lib/dpkg/), are you root
```

nach dem Versuch, irgend eine andere `apt-get`-Methode auszuführen als `source`, haben Sie keine Root-Rechte, d.h. Sie haben sie als normaler Benutzer ausgeführt.

Der gleiche Fehler wie oben tritt auf, wenn versucht wird, zweimal `apt-get` gleichzeitig auszuführen oder auch, wenn versucht wird `apt-get` auszuführen während ein `dpkg` Prozess läuft. Die einzige Methode, die simultan zu anderen ausgeführt werden darf, ist die `source`-Methode.

Wenn eine Installation mitten im Prozess abbricht und Sie merken, dass es nicht mehr möglich ist, Pakete zu installieren oder zu entfernen, versuchen Sie diese zwei Befehle auszuführen:

```
# apt-get -f install
# dpkg --configure -a
```

Danach versuchen Sie es erneut. Es kann nötig sein, den zweiten der beiden Befehle mehr als einmal auszuführen. Dies ist eine wichtige Information für die Abenteurer, die »unstable« benutzen.

Tritt »E: Dynamic MMap ran out of room« beim Ausführen von `apt-get update` auf, sollte die folgende Zeile der `/etc/apt/apt.conf` hinzugefügt werden:

```
APT::Cache-Limit 10000000;
```

## 11.2 Wo gibt es Hilfe?

Wenn Sie sich von Zweifeln geplagt fühlen, ziehen Sie die umfangreiche Dokumentation des Debian-Paketsystems zu Rate. `--help` und Manpages können eine enorme Hilfe sein, genau wie die Dokumentation in den Verzeichnissen in `/usr/share/doc` wie `/usr/share/doc/apt`.

Wenn diese Dokumentation nicht ausreicht, um Ihre Probleme zu beseitigen, versuchen Sie es auf den Debian-Mailinglisten. Mehr Informationen über die speziellen Benutzer-Listen gibt es auf der Debian-Webseite: `http://www.debian.org`.

Natürlich sind diese Listen und Hilfen nur für Debian-Benutzer; Benutzer anderer Systeme werden von der Gemeinschaft Ihrer Distribution bessere Hilfe erlangen.

# Kapitel 12

# Welche Distributionen unterstützen APT?

Hier finden Sie die Namen einiger Distributionen, die APT unterstützen:

Debian GNU/Linux (http://www.debian.org) - Für diese Distribution wurde APT entwickelt

Mandriva Conectiva (http://www.conectiva.com.br) - Die erste Distribution, die APT mit RPM benutzte

Libranet (http://www.libranet.com)

Mandriva (http://www.mandriva.com (http://www.mandriva.com/))

PLD (http://www.pld.org.pl)

Vine (http://www.vinelinux.org)

APT4RPM (http://apt4rpm.sf.net)

Alt Linux (http://www.altlinux.ru/)

Red Hat (http://www.redhat.com/)

Sun Solaris (http://www.sun.com/)

SuSE (http://www.suse.de/)

Ubuntu (http://www.ubuntulinux.com/)

Yellow Dog Linux (http://www.yellowdoglinux.com/)

# Kapitel 13

# Danksagungen

Ein großes Dankeschön geht an meine Freunde aus dem Debian-BR-Projekt und Debian selbst, die eine ständige Hilfe für mich sind und mir immer die Kraft geben, für das Wohl der Menschheit zu arbeiten genauso wie Sie mir bei meinem Ziel helfen, die Welt zu retten. :)

Außerdem möchte ich CIPSGA für die enorme Hilfe danken, die es für unser Projekt und alle freien Projekte die aus großen Ideen entstehen bedeutet.

Und spezieller Dank geht an:

Yooseong Yang <yooseong@debian.org>

Michael Bramer <grisu@debian.org>

Bryan Stillwell <bryan@bokeoa.com>

Pawel Tecza <pawel.tecza@poczta.fm>

Hugo Mora <h.mora@melix.com.mx>

Luca Monducci <luca.mo@tiscali.it>

Tomohiro KUBOTA <kubota@debian.org>

Pablo Lorenzzoni <spectra@debian.org>

Steve Langasek <vorlon@netexpress.net>

Arnaldo Carvalho de Melo <acme@conectiva.com.br>

Erik Rossen <rossen@freesurf.ch>

Ross Boylan <RossBoylan@stanfordalumni.org>

Matt Kraai <kraai@debian.org>

Aaron M. Ucko <ucko@debian.org>

Jon Åslund <d98-jas@nada.kth.se>

Isaac Jones <ijones@syntaxpolice.org>

# Kapitel 14

# Neue Versionen dieses Dokuments

Diese Anleitung wurde vom Debian-BR (http://www.debian-br.org/)-Projekt mit dem Ziel geschrieben, den täglichen Umgang mit Debian zu erleichtern.

Neue Versionen dieses Dokuments werden auf der Webseite des Debian-Dokumentationsprojekts zur Verfügung gestellt werden http://www.debian.org/doc/ddp.

Kommentare und Kritik können direkt per Mail an mich gesandt werden unter der Adresse <kov@debian.org> oder als Fehler im Debian BTS gegen das Paket apt-howto eingereicht werden.

Kritik zur deutschen Übersetzung geht an David Spreen <netzwurm@debian.org> und Florian »eix« Rehnisch <eixman@gmx.de>.